Security Target

for

SecDocs Security Komponenten Version 2.4

Version	Date	Author	Comments
0.1	25.05.2015	Stefan Dörpinghaus	Initial version of updated ST v.1.0
0.2	15.07.2015	Stefan Dörpinghaus	Updated version due to results from a workshop with the certification and evaluation body
0.3	05.08.2015	Stefan Dörpinghaus	Updated version due to internal comments
0.4	30.09.2015	Stefan Dörpinghaus	Final version ready for evaluation
0.5	16.11.2015	Stefan Dörpinghaus	Corrections according to observations made by the evaluation body
0.6	17.01.2016	Stefan Dörpinghaus	Corrections according to further comments from the evaluation body
0.7	29.01.2016	Stefan Dörpinghaus	Corrections according to further comments from the evaluation body
0.8	31.08.2016	Stefan Dörpinghaus	Corrections according to further comments from the certification facility
0.9	05.09.2016	Stefan Dörpinghaus	Correction of formal error in FPT_TDC.1.1
1.0	16.09.2016	Stefan Dörpinghaus	Correction of inappropriate descriptions in the rationale
1.1	22.01.2016	Stefan Dörpinghaus	Typo fixed
1.2	16.02.2018	Stefan Dörpinghaus	Installation Tool omitted; "*Profil-ID*" renamed to "*ProfileIdentToken*"
1.3	09.08.2018	Stefan Dörpinghaus	Corrections according to AGD-OR v.1.3
1.4	11.10.2018	Stefan Dörpinghaus	Amendment of Appendix A
1.5	12.12.2018	Stefan Dörpinghaus	Listing the hash values of the TOE components
1.6	01.04.2019	Stefan Dörpinghaus	Listing the SHA-256 hash values of the TOE components; description of last TOE changes
1.7	16.05.2019	Andreas Menke	Changes according to ZK_0994_ASE_V1.4
1.8	21.05.2019	Andreas Menke	Clarifications according to ZK_0994_ASE_V1.4
1.9	29.05.2019	Stefan Dörpinghaus	Changes according to ZK_0994_ASE_V1.7
2.0	04.06.2019	Stefan Dörpinghaus	Changes according to comments to 0994_OR_AGD_v1_6
2.1	06.06.2019	Stefan Dörpinghaus	Changes according to comments to 0994_OR_AGD_v1_7
2.2	26.06.2019	Stefan Dörpinghaus	Changes according to ZK_0994_ASE_V1.9
2.3	02.07.2019	Stefan Dörpinghaus	Changes according to ZK_0994_ASE_V1.10
2.4	08.07.2019	Stefan Dörpinghaus	Changes according to ZK_0994_ASE_V1.12
2.5	22.07.2019	Stefan Dörpinghaus	Formal correction in chapter 7.3
2.6	06.08.2019	Stefan Dörpinghaus	Final version
2.7	23.08.2019	Stefan Dörpinghaus	Final version

Contents

List of Figures

List of Tables

1 ST Introduction

This document represents a Security Target (ST) for the software *SecDocs Security Komponenten Version 2.4* enabling the long-term preservation of electronic documents by implementing the ArchiSafe concept developed by the Physikalisch-Technische Bundesanstalt (PTB) - the German National Metrology Institute providing scientific and technical services.

1.1 ST Reference

ST Name:	Security Target for *SecDocs Security Komponenten Version 2.4*
TOE:	SecDocs Security Komponenten Version 2.4
Certification ID:	BSI-DSZ-CC-0994
ST Version:	2.7
Date:	23.08.2019
Sponsor:	OpenLimit SignCubes AG
Editors:	OpenLimit SignCubes AG
CC Version:	3.1 (Revision 4)

This document contains the Security Target of the software *SecDocs Security Komponenten Version 2.4* which is from now on called TOE ("*Target of Evaluation*").

This Security Target is compliant to the Common Criteria Protection Profile for an "ArchiSafe Compliant Middleware for Enabling the Long-Term Preservation of Electronic Documents" (BSI-CC-PP-0049-2014) [5].

1.2 TOE Reference

The TOE described in this ST is the software named "SecDocs Security Komponenten Version 2.4" manufactured by the OpenLimit SignCubes AG. The TOE is part of the software product "SecDocs Version 3.0", but can also be utilized in other software products. The TOE's components and services are described in the following chapter. The software product "SecDocs Version 3.0" comprises beside

the TOE a Client Software Application (CS), a Crypto Provider component and Storage Plugins offering the possibility to attach Storage Systems to the product. Herein, the CS allows the web-based connection of customers delivering data to the CS, and the Crypto Provider component offers all the necessary cryptographic services including the access the external timestamp providers.

1.3 TOE Overview

Legally compliant electronic business based on electronic documents is not possible without serious precautions to ensure the authenticity and integrity of the digital information, at least for the time schedule of legally specified and regulated retention times. The ArchiSafe approach to long-term preservation of evidence of (cryptographically signed) electronic documents claims:

- to use permanent and standardized document formats for the contents data only, which guarantees the long-term readability of the stored information,
- to package the contents data together with all the business information, required for a complete reconstruction of the business operation in the future in a self-contained archive object,
- to protect the evidential integrity and authenticity of the actual content (primary information) by strong cryptographic operations, like digital signatures and digital time-stamps,
- to sustain the non-repudiation of cryptographically signed and archived information objects by evidential proof and renewal of the electronic signatures,
- to reduce the dependencies from obsolescent IT infrastructure and storage technology by a straight service-oriented, multi-tier and client capable architecture.

The TOE specified in this ST enforces decoupling and access control storage systems used for the long-term preservation of (cryptographically signed) electronic documents. The TOE also enforces the provisioning of a justification, if archived data shall be deleted before its retention time.

1.3.1 Usage and major security features of the TOE

The target of evaluation (TOE) is a software product providing amongst others the core of an ArchiSafe compliant archive middleware which acts as security gateway to storage solutions. The TOE mainly decouples the data flow (i.e. the flow of data objects to be archived) between third party applications, such as document management systems, and the storage solutions. The architecture of the complete system is exemplarily shown in Figure 1.

The **client software application (CS)** submits the (cryptographically signed) information to be preserved in a **submission information package (SIP)**[1] [2] to the **storage unit (SU)** via the TOE. The TOE identifies and authenticates the requesting CS and manages the verification of the submission information packages for compliance to rules defined by the administrator of the TOE.[3] This includes the management of checks concerning the existence, the quality and the validity of the digital signatures potentially contained in the submission information package or the execution of cryptographic operations like creation of signatures or timestamps for sealing (unsigned) data before depositing them in the storage. For cryptographic operations the TOE interfaces an external Crypto Provider, denominated as **Crypto-Module** in Figure 1.

The storage unit in the back-end receives the submitted submission information package from the TOE for saving. The stored data object is now called **archival information package (AIP)**.

1 The denomination follows the OAIS framework for sharing archival notions. OAIS distinguishes between what is preserved, an Archival Information Package (OAIS AIP), what is submitted to the archive, a Submission Information Package (OAIS SIP), and what is delivered to the archive clients, a Dissemination Information Package (OAIS DIP), s. also: http://www.personal.leeds.ac.uk/~ecldh/cedars/ieee00.html Deviating from OAIS framework and for reasons of better distinctness this document uses the denomination Submission Information Package for all information packages to be archived which will be submitted from a client software application via the TOE to a storage unit. Vice versa all information packages stored in a storage unit which can be requested by client software application are denominated as Archival Information Packages.

2 For the clarification of the usage of the term “SIP”, it is important to say that an information package sent to the TOE for being archived which is transferred between the CS and the TOE or between the TOE and the Crypto-Module is called “SIP” in the context of this document.

3 See definition of “verification” in chapter 3.1.3.

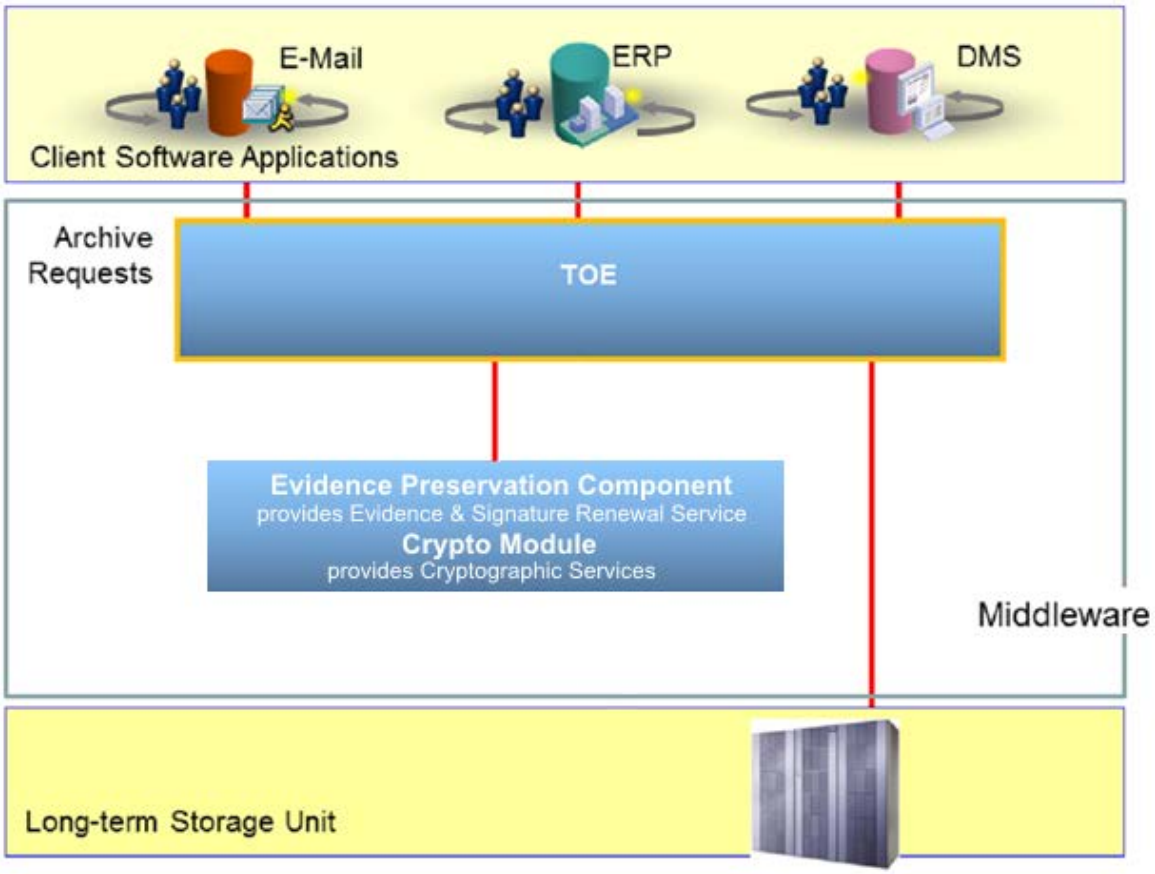


Figure 1: Architectural Overview

The TOE quits the successful storage of the AIP by sending back a unique **archive object identifier (AOID)** to the requesting CS. This AOID may be generated outside the TOE, e.g. by the storage unit or by another non-TOE part of the middleware and is required for accessing the archive information package in the future by the CS.

The trustworthy and non-TOE Evidence Preservation Component in Figure 1 manages the execution of necessary functionalities and/or mechanisms to preserve the integrity, authenticity and nonrepudiation of the saved data. The cryptographic operations needed by the Evidence Preservation Component are implemented by the Evidence Preservation Component itself because it acts as the TOE's Crypto-Module, too.

Based on the functionality to decouple the data flow between client applications and the storage systems, the TOE provides the following general security functionalities:

(SF 1) preventing the access to the storage systems from unknown client applications by reliable identification and authentication of these external entities,

(SF 2) preventing the storage of submission information packages (SIP) which in whole or in part cannot be verified successfully corresponding to the rules deposited in the TOE in order to guarantee interoperability between client applications and storage systems,

(SF 3) forwarding of successfully verified SIP's to the dedicated storage systems only or another trusted application which in turn forwards the SIP to the dedicated storage systems only,

(SF 4) preventing the deletion of AIP's before the expiry of their retention time without a justification,

(SF 5) retrieval and delivery of AIP from the dedicated storage system (to the CS) only

The TOE itself does not provide any mechanisms for the preservation of the integrity, authenticity and non-repudiation of (cryptographically signed) electronic documents by creation, proof or renewal of evidence data or data relevant to evidence, like electronic signatures or timestamps. The TOE does also not protect the confidentiality of the documents.

1.3.2 TOE Type

The TOE is a software library being part of a TR-03125 compliant[4] IT middleware component that trustworthy and reliable mediates and controls the access to a SU for submission of SIP's, retrieval or deletion of AIP's or requests of evidence records of AIP's.

The TOE consists of a set of jar archives each representing one of the following three TOE components (*MigSafe*, *OverSign*, *CredentialStore*). The TOE component *MigSafe* constitutes the base of security gateway controlling the access of business applications to the storage systems. The TOE component *OverSign* acts as a gateway to the Evidence Preservation Component and offers infrastructural methods for building the data structures needed for the generation and renewal of

[4] The TOE manufacturer is planning a TR-03125 compliance certification.

evidence records proving the unmodified existence of archival information packages at a certain time. The TOE component *CredentialStore* is used for the management and storage of user accounts and their profiles during runtime of the TOE.

For being used the TOE has to been integrated in a software component by the TOE integrator. In other terms the TOE can only been run in its integrated form. The software product "*SecDocs v.3.0*" offers such an integrated form of the TOE.

1.3.3 Required non-TOE hardware / software

Operational Environment

The TOE runs as an application on an IT system and needs the protection by the underlying system platform, e.g. the operating system. The TOE is intended to be run at least in a protected environment specified in [20] as "geschützter Einsatzbereich". The TOE is intended to be integrated in an IT product running on servers supported by the TOE. The machine running this server must at least have a 2 GHz processor, 4 GB RAM, and 200 MB hard disk space.

The IT environment of the TOE is protected by virus and malware protection components and the underlying system platform is protected against network based attacks. It protects the TOE and the resources used by the TOE against unauthorized modifications by suitable access protection mechanisms. These resources needed by the TOE include the configuration data needed for the TOE startup process. As described in the TOE manuals, the trustworthy TOE administrator has to make sure that the appropriate configuration is loaded into the TOE. Additionally, the log data generated by the TOE are protected by the underlying IT environment and the operational environment of the TOE and these data are appropriately handled by the trustworthy TOE administrator.

The TOE itself does not implement any cryptographic mechanisms for protecting or evaluating the integrity and authenticity of the data to be saved. For this purpose the TOE uses trustworthy crypto providers which are explicitly not part of the TOE. For the communication between the TOE and the *Crypto Provider Component* "OpenLimit Middleware Version 3 Server" a socket-based communication according to [21] is used.

User and administrators of the product are trustworthy and follow the instructions of the user guidance delivered with the TOE.

Operating System

The TOE runs as part of an application on an IT system and needs the protection by the underlying system platform, e.g. the operating system plus a Java Virtual Machine.

For being used the TOE needs to be integrated in an ArchiSafe compliant archive middleware. Therefore, the TOE needs further parts of the ArchiSafe architecture being supplied by the TOE integrator: the TOE needs an implementation of a *Client Software Application* (CS), of the *Crypto Provider Component*, of the *Evidence Preservation Component* and of a so-called *Storage Plugin* as a trustworthy application interfacing with the long-term storage system (SU). The TOE is designed for the usage of the "OpenLimit Middleware Version 3 Server" available from OpenLimit SignCubes AG. This product acts as the TOE's *Crypto Provider Component* and *Evidence Preservation Component*.

The *Client Software Application*, the *Crypto Provider Component*, the *Evidence Preservation Component*, the *Storage Plugin* and the *Storage Unit* (or another trustworthy applications interfacing with the SU) - as further parts of the ArchiSafe architecture - are not part of the TOE although the TOE depends on some features of these parties, e.g. the generation of the unique AOID by the SU (or another non-TOE part of the archive middleware).

For the integration of the TOE in an ArchiSafe compliant middleware, *OpenJDK 8 Update 202* in its 64 bit version for Linux is needed.[5]

For the execution of the TOE being integrated in an ArchiSafe compliant archive middleware acting as a secure archive gateway, the so-called "integrated form" of the TOE, the runtime environment of *OpenJDK 8 Update 202* in its 64 bit version for Linux is needed.[6]

5 Please consider the information about updating the Java version in the integrator's manual *MSOS_Integratorhandbuch_DE.pdf*, chapter 2.1.3.

6 Please consider the information about updating the Java version in the administrator's manual *MSOS_Admininstratorhandbuch_DE.pdf*, chapter 2.3.

The following operating systems (in their 64-bit version for AMD64/x64) are supported by the TOE:

- RedHat Enterprise Linux RHEL 6.5
- RedHat Enterprise Linux RHEL 7.0
- SUSE Linux Enterprise Server 11 SP 3

1.3.4 Scope of the TOE

The TOE is included in the archive file *SecDocs_Security_Components_V.2.4.zip* which is qualified electronically signed by the manufacturer's senior IT security consultant, Mr. Stefan Dörpinghaus (1ce2ab, D-TRUST Limited Basic CA 1-4 2015) with the certificate's serial number DTRWM853732015702322. The RSA-2048 public key of this certificate is

30 82 01 0a 02 82 01 01 00 8c d4 73 d1 5d be 42 57 73 84 93 91 e2 61 46 aa 54 01 99 a3 52 f1 a0 dd d0 36 e5 1b 96 0e 14 f6 99 76 89 a8 e0 31 0f 98 ca ea 4a c7 c7 20 83 b3 02 c8 4f 9f b4 18 b6 ab 64 6f f7 98 90 44 27 c3 3c 7b c3 19 65 fa b2 a3 0b 0b 16 e9 e9 ce 82 ea c5 16 09 e1 1d 74 7c 31 77 f9 51 87 6a 18 2a 43 54 a8 6b 76 4f 4a 07 f2 17 96 3b ab 40 4c b5 6e 92 55 57 f4 c1 7a 02 bc 65 62 c0 94 63 e2 7c af 42 87 64 5c 52 d6 65 ff 87 a5 7b 10 8a 92 0d b6 cf fb c8 5a 75 73 a9 12 87 eb 6c 55 34 cf 05 16 0f 31 d1 c1 26 93 23 4c 6b 86 22 15 bc f9 77 88 7b d9 6e 9b 54 b7 1b b8 d6 f3 d5 1d 24 16 11 ba f5 ed da 9e 67 14 31 4a fa c2 e9 dc 2e 15 e0 2d 6f ce bc cb 82 1c 1f 77 e1 e3 a1 01 2f c1 92 2e a5 b8 1f 26 2b 25 9e dd a7 8b 9b 42 a4 15 3b cd e8 fc ed 92 dd bf 05 bf 8b 27 52 f6 a9 8c 6f f5 02 03 01 00 01

The SHA-256 hash value of the archive file *SecDocs_Security_Components_V.2.4.zip* is EEF9A24B2C31330624ADCD5A8CD83A7A145469F59E1301302FB4CB22F76271A4.

The archive file *SecDocs_Security_Components_V.2.4.zip* itself comprises the following parts:

- the archive file *MigSafeOverSign-V2.4.zip* containing the TOE's JAR files and the JavaDoc API documentation,
- the integrator's manual *MSOS_Integratorhandbuch_DE.pdf*,
- the administrator's manual *MSOS_Admininstratorhandbuch_DE.pdf*,
- the administrator's manual of the non-TOE component *OpenLimit Middleware Version 3 Server, Produktversion 1.6* [23],
- the archive file *MigSafeOverSign-V2.4.563_13022_SecureInterfaceTools.zip* containing examples for the TOE integration, and
- the TOE's functional specification.

TOE part	Name of the TOE part	SHA-256 hash value
TOE library *MigSafe*	MigSafeLibrary.jar	794E232C25735C6666F7A73605CB03F0 9E12832DFD949C28FCC09012E83A4DCF
TOE library *OverSign*	OverSignLibrary.jar	A1568D5CE04535CC27C6E3D355D33456 66935F538F201F1B501322A38E731083
TOE library *CredentialStore*	CredentialStore.zip	CDF2BE3505229716E4305BD01DB43E10 7D2E14CBCF4161EE67BAEE4A064A724A
TOE library *Filter*	XMLFilter.jar	8009712E6387052D3F9D2B22263B2AB1 15A22C54B84E0C6E48021150DE8B4F11
TOE library *Bouncy Castle*	bcprov-jdk15on-160.jar	D65BF7E1A3DAE9A8AE2AD9CB64EF443E A089B1AD930DCA999F70BBAB56D9F349
TOE integrator manual	MSOS_Integratorhandbuch_DE.pdf	3AC72F1238A3905A4DE9B761E064A8D7 A150AA211B4A389FE66AEAA63BC9CD26
TOE administrator manual	MSOS_Administratorhandbuch_DE.pdf	0F26F5732266AC8FBD974676E9CD8609 98AAA873097303DA5D1E41D77AE6952D
administrator manual of OpenLimit Middleware Version 3 Server	Administratorhandbuch_V3_Server_v.1.35	02ECF718739C531F75CFB865B3B6AD82 A3A21D42FB92B1234C39A3CB74596AD8
TOE functional specification	ADV_FSP-MigSafe-OverSign_2.4_2019-08-06.pdf	8B8AA1E4EE9F6AC305DA8000D5EC927B 773A581DBCF1CC5E046942646DD45237

Table 1: Physical parts of the TOE

Included in the physical parts of the TOE listed above are the Java classes, the TOE uses for the following cryptographic operations: random number generation, AES encryption and decryption, digesting algorithm calculation, ephemeral ECC key generation, and X.63 key derivation function.

The logical scope of the TOE is defined through the following services the TOE provides:

- The TOE accepts archive requests from authenticated Client Software Applications (CS). Thus a successfully authenticated CS is allowed to
  - submit a submission information package to the storage,
  - retrieve an archive object from the storage,
  - delete an archive object within the storage,
  - request for evidence of a particular archive object and,
  - read some meta-information.
- The TOE provides an interface for so-called “Storage Plugins” as a trusted application which in turn submits the data objects to the dedicated storage system. Implementations of this interface supplied by the TOE integrator reflect the characteristics of the underlying storage system.

Chapter 1.3 “TOE Overview” and especially chapter 1.3.1 “Usage and major security features of the TOE” offer a description of the TOE’s logical security features.

2 Conformance Claims

2.1 CC Conformance Claim

This Security Target is based upon the following:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 4, CCMB-2012-09-001 [1],
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; Version 3.1, Revision 4, CCMB-2012-09-002 [2], and
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; Version 3.1, Revision 4, CCMB-2012-09-003 [3].

This Security Target claims the following CC conformance:

- Part 2 extended (the component definition is part of this Security Target)
- Part 3 conformant
- Evaluation Assurance Level (EAL) 4 + augmented with ALC_FLR.1

2.2 PP Claim / Conformance Statement

This security target claims strict conformance to the Protection Profile for an “ArchiSafe Compliant Middleware for Enabling the Long-Term Preservation of Electronic Documents” (BSI-CC-PP-0049) [5].

2.3 Package Claim

This security target is conforming to assurance package EAL4 augmented with ALC_FLR.1 defined in CC part 3 [3].

3 Security Problem Definition

3.1 Definitions

3.1.1 Subjects

Administrator (Admin)

The **Administrator** installs the TOE and is in charge of the correct configuration of the TOE. In particular the Administrator is responsible for the correct implementation of the rules needed for a verification of submission information packages.

Another trustworthy application

This term is usually equivalent with Evidence Preservation Component in Figure 1 but can also identify any other trustworthy external i.e. non-TOE component which is interconnected between the TOE and the storage unit and provides an interface to the TOE equivalent with the storage interface.

Client

An agency or company who operates at least one CS.

Client Software Application (CS)

An external IT entity which is acting on behalf of an authorized user and capable and authorized to use the TOE for submitting archive requests to the SU.

Crypto-Module (also called Crypto Provider)

A trusted external i. e. non-TOE component which will be used by the TOE and other non-TOE components to perform trustworthy cryptographic operations.

Evidence Preservation Component

A trustworthy external, i.e. a non-TOE, component which provides or manages any functionality and/or mechanisms to preserve the integrity, authenticity and non-repudiation of the saved data and to renew security measures which serve for the preservation of the integrity, authenticity and the non-repudiation of the saved data.

Organization using the TOE

An agency or company who operates and/or uses the TOE. It may be possible that the clients and their applications and/or the storage unit(s) are owned by another agency/company but this will not be differentiated in this ST.

Storage System or Storage Unit (SU)

A storage system which stores data for a long-term.

3.1.2 Objects

Primary Information

The contents data (primary information) representing the business information to be stored.

Application note: This ST does not want to specify the data structure or format of primary information submitted to the archive. However, it is strongly recommended to use standard formats like ASCII, PDF/A or TIFF. In case of XML-based submission information packages the primary information may be converted into a native text format (MIME Base64 coded) for embedding it in XML.

Meta Information (Metadata)

Data associated with primary information in the submission information package serving for the identification and reconstruction of the business and archive context of the primary information.

Cryptographic data relevant to evidence

Data like cryptographic signatures, certificates or any other cryptographic data which serve to assure the integrity and authenticity of data to be archived. This cryptographic data relevant to evidence is also stored in the submission information package.

Submission Information Packages (SIP's)

A conceptual data container which may comprise primary information, metadata and cryptographic data relevant to evidence, required for an evidentiary reconstruction of business transactions in the future. Submission information packages will be denominated as archival information packages when they are saved in the storage system.

Application note: This ST does not want to specify data structures of a submission information package in detail. Product developers shall be free to specify data structures of submission information packages which can be successfully verified and/or processed by their own procedures and rules deposited in the TOE.

Archival Information Packages (AIP's)

Once a submission information package was successfully checked and processed by the TOE and delivered to the SU, it is called archival information package (AIP). Archival information packages contain all primary information, metadata and cryptographic data relevant to evidence, required for an evidentiary reconstruction of business transactions in the future stored in the specified format. Archival information packages can be accessed by a uniquely identified and authorized CS only which provides a valid AOID.

Application note: This ST does not want to specify data structures of an archival information package in detail. Product developers shall be free to specify data structures of the stored archival information packages. Due to necessary preservation measures however, relating to legally prescribed retention times, it is strongly recommended to use self-contained data structures which might be verified and/or processed by rules deposited in the TOE for any retrieval request. In addition, archival information packages may be augmented with a reference to the submitting CS (e. g. stored as meta information by the TOE during the ingest).

Archive Objects

Archive Objects is the generic term for submission information packages, archive information packages, cryptographic data relevant to evidence or particular data which will be read from chosen archival information packages.

TOE configuration data

TOE internal data required for the correct execution of the security functionalities, especially for the correct and reliable identification and authentication of other units which are not part of the TOE as well as for verification of SIP's and processing of archive requests.

Rules

The rules are part of the TOE configuration data and specify operations the TOE must perform on archive objects and archive requests. Rules must be specified by the organization using the TOE.

Application note: The rules may specify that the TOE

- *must initiate to digitally sign or timestamp any submission information package[7].*
- *has to start the generation of an evidence record for any or a particular request for retrieval of archival information packages. For this purpose, the TOE may interface to an external crypto provider or to another special and trustworthy application.*

Protocol Data

Log information produced by the TOE.

Evidence Data

According to the specification of the IETF [8] cryptographic data for all AIP's calculated and maintained in order to be able to prove the integrity and authenticity of archival information packages at and since a certain time. Evidence Data as specified by the IETF are generated and maintained outside the TOE. Evidence Data are generated and/or retrieved on request as an Evidence Record for a certain AIP.

7 In cases, for example, that unsigned data shall be saved or added to the archive, cryptographic operations performed on the data may serve as a proof about the availability of the data at a certain time.

3.1.3 Operations

Archive Requests

An archive request is a call from the Client Software Application to the TOE to perform a certain operation on the storage unit. The following Archive Requests are supported by the TOE:

- **Archive Submission Request** means, the Client Software Application wants to store (new) submission information packages into the storage unit. The submission information packages are included in this archive request.
- **Archive Retrieval Request** means, a Client Software Application wants to read archival information packages from the storage unit. The retrieval request shall return the archival information packages in self-contained, open and standardized data structures and formats agreed between the organization using the TOE and the organization which operates the storage unit. Modification of the archive information packages during the retrieval is not possible.
- **Archive Deletion Request** means, the Client Software Application wants to delete particular archival information packages from the storage unit. A deletion request may happen before or after the retention time of the archival information package. The TOE enforces a justification, if archival information packages shall be deleted before expiration of the retention time.
- **Archive Evidence Request** means, the Client Software Application requests evidence data to the fact that the archival information packages exist unmodified within the storage unit since a certain point of time until now.

Verification of archive objects

Verifications of archive objects mean that the TOE enforces the processing of the archive objects in accordance with a set of rules stored in the configuration data of the TOE. This may include managing the execution of cryptographic operations which checks the validity of potentially existing digital signatures, the execution of cryptographic operations which serve

for protecting the integrity and authenticity of archive objects or renewing evidence data which prove the unmodified existence of archival information packages in the storage.

Archive Submission Request	See “Archive Requests”
Archive Retrieval Request	See “Archive Requests”
Archive Deletion Request	See “Archive Requests”
Archive Evidence Request	See “Archive Requests”

3.1.4 Security Attributes

Client Software Application Identity

All Client Software Applications which use the TOE shall have a unique identity, e.g. a numeric value or a unique name. The TOE shall connect to client software applications only whose identity is known by the TOE.

Crypto Provider Identity

Any crypto provider (denominated as Crypto-Module in Figure 1) connected to the TOE and used for performing cryptographic operations shall have a unique identity, e.g. a numeric value or a unique name. The TOE shall connect to crypto providers only whose identity is known by the TOE.

Application note: It is worth to note, that the TOE itself is required to implement secure cryptographic operations or other measures needed to provide assured identification of its endpoints and to establish a trusted channel between itself and the crypto provider. It is not assumed that the environment of the TOE will provide this channel.

Storage Unit Identity / Trustworthy Application Identity

Each storage unit connected to the TOE or another trustworthy application which in turn connects to the storage unit (e. g. the Evidence Preservation Component in Figure 1) must have a unique identifier, e.g. a numeric value or a unique name. The TOE shall only connect to storage units/trustworthy applications whose identity is known by the TOE.

Retention Time

The retention time of an archival information package is an optional attribute storing the date and time when this AIP can be deleted without justification. The value will be specified for each archival information package.

Justification

In case of an archive deletion request before end of the retention time a justification must be given documenting the reason for that premature deletion. That can be done by a free text field or selection boxes or other means.

Archive Object ID (AOID)

The archive object ID is a unique identifier of any archival information package stored in the storage unit. This AOID will be generated outside the TOE, e.g. by the storage unit or by a non-TOE part of the middleware, when a submission information package will be sent to the TOE and stored in the SU. This AOID will be returned to the submitting client software application by the TOE for using it as a security attribute for accessing the archival information package.

Archive Object Specific Credentials

The archive object ID (AOID) and the retention time and in case of an archive deletion request before end of the retention time a justification.

Another trustworthy Application Identity

All another trustworthy applications which are used by the TOE shall have a unique identity, e.g. a numeric value or a unique name. The TOE shall connect to other trustworthy applications only, if those identities are known by the TOE.

3.2 Assets

Protocol Data

The **availability** of log information generated by the TOE, provided by the TOE for usage through the TOE's Administrator, and usually stored in the storage unit has to be maintained by the TOE [8].

Configuration data stored in the Credential Store component

On start-up or during operation the Administrator must load credential data and configuration data assigned to the supported external entities into the Credential Store component of the TOE. The protection of the **integrity** of these data ensures the correct functionality to process the archive objects resp. the behaviour of the TOE to identify and authenticate these external entities.

Submission information package

The **integrity** of the submission information package (SIP) has to be maintained by the TOE.

Archival information package

The **integrity** of the archival information package (AIP) has to be maintained by the TOE.

3.3 Assumptions

The description of assumptions illustrates the security aspects of the environment in which the TOE is intended to be used.

A.ADMIN

The administrators of the TOE, of the crypto provider or other trustworthy 3rd party components connected to the TOE, of the storage system, the underlying systems, of the communication connections (e.g. the LAN) are not careless, wilfully negligent, or hostile, and will follow and abide the instructions provided by the administrator's guidance. They are well

8 It's in the responsibility of the integrator of the TOE to provide a mechanism to store the protocol data if needed.

trained to securely and trustworthy administer all aspects of TOE operation as well as all other involved processes or operations in accordance with the guidance. The administrators will protect their credentials used for authentication. Credentials must not be disclosed to other individual.

A.AUTHENT

All CS, SU, and any trustworthy special applications (e.g. the Evidence Preservation Component in Figure 1) which are authorized by the IT environment for using the TOE or to be used by the TOE, identify and authenticate the TOE before data transfer.

A.COMMUNICATION

The communication interconnections between the TOE and all non-TOE components and systems, are protected by the environment – by physical or logical security measures – against disclosure as appropriate regarding the need for information disclosure of the clients.

A.CONFIGURATION

The TOE is securely configured and all data required for the configuration operation of the TOE are secure and reliable transported to and installed on the machine which runs the TOE.

A.EVIDENCEDATA

The generation, storage, management and renewal of evidence data for proving the unmodified existence of archival information packages at a certain time will be provided by trustworthy special applications (e.g. the Evidence Preservation Component in Figure 1) in a secure non-TOE environment.

A.NO_BYPASS

The TOE is integrated in the IT environment in such a way that all storage access by the CS cannot bypass the TOE, if it is mandated or required by policies of the organization which uses the TOE.

A.PHYSPROT

The machine on which the TOE runs is protected against unauthorized physical access and modification.

A.RULES

Rules defined for operating on archive objects and archive requests by the TOE do not introduce any security risk.

A.SERVER

The machine on which the TOE, systems and applications run is free from malware and viruses. Systems and applications running on the server are securely installed. An unauthorized access to functions, processes and data of the TOE is prevented by the security mechanisms of the underlying system.

A.STORAGE

The dedicated SU provides a reliable, secure and available storage of archival information packages (AIP), even for long-terms.

A.TIMESTAMP

The environment of the TOE is able to provide reliable time-stamps to the TOE.

A.TOKEN

The environment, e. g. the SU or another non-TOE part of the middleware, provides a reliably generated unique archive object identifier (AOID) for any successfully archived submission information package.

A.TRUSTAPP

The archive requesting CS is secure, and provides reliable measures regarding the authentication and access authorization of its (human) users.

A.TRUSTCRYPTO

Only trustworthy cryptographic components are used. The cryptographic components do not send any security relevant and confidential data to any external entity and will reliably protect all security relevant and confidential data from disclosure by an external entity.

3.4 Threats

The threat agents can be categorized as either

- Unidentified individuals or client software applications, i.e. entities not known by the TOE but having access to the communication interfaces exposed by the TOE or to the client software applications, or
- Identified users of the TOE, i.e. individuals or entities, which may access resources controlled by the TOE.

The threat agents are assumed to originate from a well-known user community in a non-hostile environment. The TOE therefore protects against threats of inadvertent or casual attempts to breach the system security. The TOE is not intended to be used in environments where protection is required against determined and hostile attacks to breach the system security at all. Resuming, the following threats need to be countered by the TOE:

T.CRYPTO_SPOOF

An attacker attempts to substitute the crypto provider or to intercept and manipulate the communication between the TOE and the crypto provider.

T.DATA_ACCESS1

An attacker attempts to gain unauthorized access to the SU by using an authorized client software application in an unintended way, e.g. by sending manipulated AOIDs.

T.DATA_ACCESS2

An attacker attempts to gain unauthorized access to the SU by spoofing external entities, e.g. by simulating an authorized client software application.

T.DATA_ACCESS3

An attacker attempts to gain unauthorized access to archive objects by exploiting requests or functionalities additionally implemented by the TOE but not specified in this ST.

T.DATA_DELETION

A (user of a) CS attempts to delete an archival information package before expiry of the retention time of the AIP without any justification.

T.DATA_MODIFY

An attacker attempts to modify an archive object in a specific manner during transmission between CS and the TOE. Objective of the attacker is that the manipulated archive object will be stored or that the CS assumes that the manipulated archive object was actually stored.

T.EVIDCOMP_SPOOF

An attacker attempts to substitute the Evidence Preservation Component or to intercept and manipulate the communication between the TOE and the Evidence Preservation Component.

T.STORAGE_SPOOF

An attacker attempts to substitute the SU or another trustworthy application which in turn is dedicated to forward the SIP to the SU or to manipulate the communication between the TOE and the SU or the other trusted applications.

T.TOE_SPOOF

An attacker attempts to feign TOE functionalities to external components like the CS or the SU.

3.5 Organizational Security Policies

P.ACCESS

The TOE has to provide at least the following operations:

- Archive Submission Request,
- Archive Retrieval Request,
- Archive Deletion Request and,
- Archive Evidence Request.

P.AOID

The TOE must not interpret or change (modify) the archive object ID.

P.CONFIGURATION

The TOE must select the right configuration data per archive request, must interpret it in a correct manner and execute the rules defined within in the configuration data in the right order.

P.RETURN

After successful storage of a submission information package the TOE has to return the archive object ID (AOID) to the requesting CS.

P.RULES

In order to decouple CS and SU the TOE has to verify Archive objects according to specified rules. The verification may be performed either in the context of a submission request or vice versa in the context of a retrieval request. When the verification fails the TOE has to react in an appropriate way.

4 Security Objectives

This section defines the security objectives for the TOE and its supporting environment. The security objectives are categorized as security objectives for the TOE or for the environment.

4.1 Security Objectives for the TOE

O.ACCESS

The TOE allows at least the following operations:

- Archive Submission Request,
- Archive Retrieval Request,
- Archive Deletion Request and,
- Archive Evidence Request.

O.AOID

The TOE must not interpret or change (modify) the archive object ID.

O.AUTH_REQUEST

The TOE shall authorize archive requests based on the authenticity of the requesting client and archive object specific credentials provided (e.g. the AOID).

O.CONFIGURATION

The TOE assures the selection and application of the appropriate configuration, interprets the configuration data in a correct manner and executes the rules defined within in the configuration data in the right order. The TOE denies an archive request, if any operation defined by the rules failed or cannot completely be executed.

O.CRYPTO_SPOOF

The TOE assures that the crypto provider cannot be substituted unnoticed.

O.DATA_EXAM

The TOE assures that either the submission information packages at the point of submission or the archival information packages at the point of retrieval request will be verified according to the specified rules.

O.DELETION

The TOE assures that archival information packages can only be deleted by client requests before expiry of the retention time when the delete request will be submitted together with a justification.

O.DELETION_LOG

The TOE must log any delete requests and the accompanying justification, if the retention time of these archive objects is not yet expired.

O.RETURN

After successful storage of submission information packages the response of the TOE to the requesting CS must contain at least the archive object IDs (AOID).

O.STORAGE_SPOOF

The TOE assures that the SU or another trustworthy application which in turn is connected to the SU and will be used for saving and retrieving the archive data objects cannot be replaced without notice (this includes especially also an Evidence Preservation Component).

O.TOE_AUTHENT

The TOE is capable to authenticate itself against external non-TOE entities.

O.TOE_COMM

The TOE shall be capable to protect the communication between itself, the CS, the SU, the crypto provider and all other trustworthy application (e. g. an Evidence Preservation Component as shown in Figure 1) against modification.

4.2 Security Objectives for the Operational Environment

OE.ADMIN

The administrators of the TOE, of the crypto provider cryptographic or other trustworthy 3rd party components connected to the TOE, of the storage system, the underlying systems, and of the communication connections (e.g. the LAN) must not be careless, wilfully negligent, or hostile, and will follow and abide the instructions provided by the administrator's guidance. They shall be well trained to securely and trustworthy administer all aspects of TOE operation as well as all other involved processes or operations in accordance with the guidance. The administrators shall protect their credentials used for authentication. Credentials must not be disclosed to other individual.

OE.AUTHENT

The client software applications (CS), the SU, and any trustworthy special applications (e.g. the Evidence Preservation Component in Figure 1) which are authorized by the IT-Environment for using the TOE or to be used by the TOE, have to be configured in such a way that they identify and authenticate the TOE before any data transfer.

OE.COMMUNICATION

The communication interconnections between the TOE and all non-TOE components and systems, have to be protected by the environment – by physical or logical security measures – against disclosure as appropriate regarding the need for information disclosure of the clients. The communication interconnections between the TOE and all non-TOE components and systems must be protected by the environment – by physical or logical security measures – against threats (e. g. disclosure) which may compromise the security objectives of this ST.

OE.CONFIGURATION

The TOE has to be securely configured and all data required for the configuration of the TOE must secure and reliable transported to and installed on the machine which runs the TOE.

OE.EVIDENCEDATA

The generation, storage, management and renewal of evidence data for proving the unmodified existence of archival information packages at a certain time shall be provided by trustworthy special applications (e.g. the Evidence Preservation Component in Figure 1) in a secure non-TOE environment.

OE.NO_BYPASS

The TOE must be integrated in the IT environment in such a way that all storage access by the CS cannot bypass the TOE, if it is mandated or required by policies of the organization which uses the TOE.

OE.PHYSPROT

The machine on which the TOE runs must be protected against unauthorized physical access and modification.

OE.RULES

Rules defined for operating on archive objects and archive requests by the TOE must not introduce any security risk.

OE.SERVER

The machine on which the TOE, systems and application run must be free from malware and viruses. Systems and applications running on the server must be securely installed. An unauthorized access to functions, processes and data of the TOE must not be possible.

OE.STORAGE

The dedicated SU must provide a reliable, secure and available storage of archival information packages (AIP), even for long-terms.

OE.TIMESTAMP

The environment shall be able to provide reliable time-stamps to the TOE.

OE.TOKEN

The environment, e. g. the SU or another non-TOE part of the middleware, has to provide a reliably generated unique archive object identifier (AOID) for any successfully archived submission information package.

OE.TRUSTAPP

The archive requesting CS has to provide sufficient trust to be assumed as secure and has at least to provide reliable measures regarding the authentication and access control of its (human) users.

OE.TRUSTCRYPTO

Only trustworthy cryptographic components are allowed to be used. The cryptographic components must not send any security relevant and confidential data to any external entity and have to reliably protect all security relevant and confidential data from disclosure by an external entity.

4.3 Rationale For Security Objectives

This chapter explains how each aspect of the security environment of the TOE will be covered by the security objectives. In addition the security environment is explained.

	O.ACCESS	O.AOID	O.AUTH_REQUEST	O.CONFIGURATION	O.CRYPTO_SPOOF	O.DATA_EXAM	O.DELETION	O.DELETION_LOG	O.RETURN	O.STORAGE_SPOOF	O.TOE_AUTHENT	O.TOE_COMM	OE.ADMIN	OE.AUTHENT	OE.COMMUNICATION	OE.CONFIGURATION	OE.EVIDENCEDATA	OE.NO_BYPASS	OE.PHYSPROT	OE.RULES	OE.SERVER	OE.STORAGE	OE.TIMESTAMP	OE.TOKEN	OE.TRUSTAPP	OE.TRUSTCRYPTO
T.CRYPTO_SPOOF					X							X														
T.DATA_ACCESS1			X						X																	
T.DATA_ACCESS2			X						X																	
T.DATA_ACCESS3	X		X															X								
T.DATA_DELETION							X	X																		
T.DATA_MODIFY											X	X		X	X											
T.EVIDCOMP_SPOOF										X		X														
T.STORAGE_SPOOF										X		X														
T.TOE_SPOOF											X			X												
P.ACCESS	X																									
P.AOID		X																								
P.CONFIGURATION				X									X			X										
P.RETURN									X																	
P.RULES						X																				
A.ADMIN													X													
A.AUTHENT														X												
A.COMMUNICATION															X											
A.CONFIGURATION																X										
A.EVIDENCEDATA																	X									
A.NO_BYPASS																		X								
A.PHYSPROT																			X							
A.RULES																				X						
A.SERVER																					X					
A.STORAGE																						X				
A.TIMESTAMP																							X			
A.TOKEN																								X		
A.TRUSTAPP																									X	
A.TRUSTCRYPTO																										X

Table 2: Coverage of the Assumptions

4.3.1 Coverage of the Assumptions

A.ADMIN: A.ADMIN is directly covered by OE.ADMIN.

A.AUTHENT: A.AUTHENT is directly covered by OE.AUTHENT.

A.COMMUNICATION: A.COMMUNICATION is directly covered by OE.COMMUNICATION.

A.CONFIGURATION: A.CONFIGURATION is directly covered by OE.CONFIGURATION.

A.EVIDENCEDATA: A.EVIDENCEDATA is directly covered by OE.EVIDENCEDATA.

A.NO_BYPASS: A.NO_BYPASS is directly covered by OE.NO_BYPASS.

A.PHYSPROT: A.PHYSPROT is directly covered by OE.PHYSPROT.

A.RULES: A.RULES is directly covered by OE.RULES.

A.SERVER: A.SERVER is directly covered by OE.SERVER.

A.STORAGE: A.STORAGE is directly covered by OE.STORAGE.

A.TIMESTAMP: A.TIMESTAMP is directly covered by OE.TIMESTAMP.

A.TOKEN: A.TOKEN is directly covered by OE.TOKEN.

A.TRUSTAPP: A.TRUSTAPP is directly covered by OE.TRUSTAPP.

A.TRUSTCRYPTO: A.TRUSTCRYPTO is directly covered by OE.TRUSTCRYPTO.

4.3.2 Encounter the Threats

T.CRYPTO_SPOOF: This threat is covered by O.CRYPTO_SPOOF (prevents spoofing of the crypto provider without notice) and O.TOE_COMM (prevents unnoticed manipulation of communication between TOE and the crypto provider).

T.DATA_ACCESS1: This threat is covered by O.AUTH_REQUEST (enforces an access control policy) and O.RETURN (ensures that only submitting CS also receives the respective AOID to be used for later access).

T.DATA_ACCESS2: This threat is covered by O.AUTH_REQUEST (enforces an access control policy) and O.RETURN (ensures that only submitting CS also receives the respective AOID to be used for later access).

T.DATA_ACCESS3: This threat is covered by O.ACCESS (specification of the core functions of the TOE), O.AUTH_REQUEST (enforces an access control policy on all functions the TOE may provide) and OE.NO_BYPASS (ensures that the TOE and its access control function cannot be bypassed by other means provided by the IT environment).

T.DATA_DELETION: This threat is directly covered by O.DELETION. In addition O.DELETION_LOG ensures that all justifications related to such delete operations will be recorded to provide evidence for correct TOE operation or for auditors.

T.DATA_MODIFY: This threat is directly covered by O.TOE_COMM. Additionally, OE.AUTHENT and O.TOE_AUTHENT enforces resp. enables a bi-directionally authentication of CS and TOE, which prevents a simple man-in-the-middle attack. OE.COMMUNICATION protects the network traffic against disclosure, which makes a directed modification more difficult.

T.EVIDCOMP_SPOOF: This threat is covered by O.STORAGE_SPOOF (prevents spoofing of an Evidence Preservation Component without notice) and O.TOE_COMM (prevents unnoticed manipulation of communication between TOE and an Evidence Preservation Component as shown in Fig.1).

T.STORAGE_SPOOF: This threat is covered by O.STORAGE_SPOOF (prevents spoofing of the storage without notice) and O.TOE_COMM (prevents unnoticed manipulation of communication between TOE and the storage).

T.TOE_SPOOF: This threat is directly covered by O.TOE_AUTHENT (enables the TOE to be authenticated by other components) and especially by OE.AUTHENT, which ensures that all the other components authenticate the TOE before any data transfer. This ensures that spoofing of the TOE would be noticed.

4.3.3 Implementation of Organizational Security Policies

P.ACCESS: This OSP is directly covered by O.ACCESS.

P.AOID: This OSP is directly covered by O.AOID.

P.CONFIGURATION: This OSP is directly covered by O.CONFIGURATION. Additionally, OE.ADMIN and OE.CONFIGURATION ensures that the TOE is correctly and securely installed and that the rules are configured as intended by the organization operating the TOE.

P.RETURN: This OSP is directly covered by O.RETURN.

P.RULES: This OSP is directly covered by O.RETURN.

5 Extended Components Definition

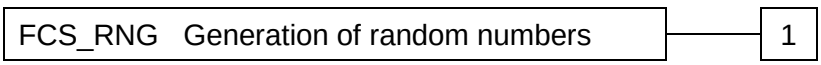
5.1 Definition of the Family FCS_RNG

This section describes the functional requirements for the generation of random number to be used as secrets for cryptographic purposes or authentication. The IT security functional requirements for a TOE are defined in an additional family (FCS_RNG) of the Class FCS (cryptographic support). The family "*Generation of random numbers (FCS_RNG)*" is specified as follows.

FCS_RNG Generation of random numbers

Family behaviour: This family defines quality requirements for the generation of random numbers that are intended to be used for cryptographic purposes.

Component levelling:



FCS_RNG.1	Generation of random numbers requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.
Management:	FCS_RNG.1 There are no management activities foreseen.
Audit:	FCS_RNG.1 There are no actions defined to be auditable.

FCS_RNG.1 Random number generation

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FCS_RNG.1.1	The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].
FCS_RNG.1.2	The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

6 Security Requirements

This section comprises security functional and security assurance requirements that shall be fulfilled by a product that is conformant to this security target.

In this section the following typographic conventions have been used:

- Selections performed have been marked in *italics*.
- Assignments performed have been marked in **bold**.
- Refinements have been marked as underlined.
- Iterations of security requirements have been marked by applying an additional identifier to the appropriate component names.

- Operations, which are not executed, are reproduced from [1], [2], and [3] without any changes.
- Uncompleted Operations are still written in brackets containing at first the executed part of the operation and subsequently the specification of the operation to be performed.

6.1 Security Policies (TSPs)

6.1.1 Access Control Policy (TSP_ACC)

The TOE shall control the access to the archive according to the following rules:

- Only identified and authenticated Client Software Applications (CS) will get permission for accessing the storage unit for writing a new SIP.
- Only securely identified and authenticated Client Software Applications (CS) which uses valid archive requests and provides archive object specific credentials will get permission for accessing the storage unit and the respective archive objects for read, delete and read evidence data.
- Only securely identified and authenticated Client Software Applications (CS) which uses valid archive requests and provide a justification will get permission to delete AIP before expiry of its retention time.

6.1.2 Information Flow Control Policy (TSP_IFC)

The TOE shall implement an information flow control policy which follows the following rules:

- All rules specified by the organization using the TOE, either at submission or at retrieval request.
- The TOE must not perform an archive request, if an operation defined by the rules deposited in the TOE cannot be performed successfully.
- The TOE shall return the archive object ID as result of a successful archive submission request.

Application note: All rules specified for archive object verification as well as potential additional rules specified by the organization using the TOE or the product developer shall be performed by the TOE in accordance with the specification and in the context of the respective archive request.

6.2 Security Functional Requirements (SFRs)

6.2.1 Class FAU: Security Audit

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the *not specified* [9] level of audit; and c) • **Successful and unsuccessful archive deletion requests for archival information packages whose retention time is not yet expired.** • **Unsuccessful authentications of Client Software Applications, Crypto Providers, the storage unit and other trustworthy applications connected to the TOE** • **Unsuccessful attempts to access Archival Information Packages[10]** • **other specifically defined auditable events: none[11].**
FAU_GEN.1.2	The TSF shall record within each audit record at least the following information a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the ST[12], • **for successful archive deletion requests to archival information packages whose retention time is not yet expired, the justification[13],** • **other audit relevant information, resulting from additional implemented requests and/or functionalities: none[14].**

9 [selection, choose one of: *minimum, basic, detailed, not specified*]

10 [assignment: *other specifically defined auditable events*]

11 [assignment: *other specifically defined auditable events*]

12 [refinement: *PP/ST*]

13 [assignment: *other audit relevant information*]

14 [assignment: *other audit relevant information*]

6.2.2 Class FCS: Cryptographic Support

6.2.2.1 Cryptographic support for TLS

FCS_CKM.1/TLS: Cryptographic key generation for TLS

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/TLS	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **TLS-PRF with SHA-256**[15] and specified cryptographic key sizes **128 bit**[16] that meet the following*:* **[9] in combination with [14] and [24]**[17].

FCS_COP.1/TLS: Cryptographic operation for TLS

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/TLS	The TSF shall perform **TLS encryption, decryption, and integrity protection**[18] in accordance with a specified cryptographic algorithm **TLS cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256**[19] and cryptographic key sizes **128 bit**[20] that meet the following: **[9], [11], [15], [25], [26] and [27]**[21].

Application note: The TOE must fulfil the requirements FCS_CKM.1/TLS and FCS_COP.1/TLS for the establishment of the communication channel between itself and a crypto provider according to FTP_ITC.1 (CRYPTO) and for the establishment of the communication channel between itself and a remote trustworthy application (e. g. the Evidence Preservation Component in Figure 1) according to

15 [assignment: *key generation algorithm*]

16 [assignment: *cryptographic key sizes*]

17 [assignment: *list of standards*]

18 [assignment: *list of cryptographic operations*]

19 [assignment: *cryptographic algorithm*]

20 [assignment: *cryptographic key sizes*]

21 [assignment: *list of standards*]

FTP_ITC.1 (TAPP). Please note, that due to the TOE's constructional constitution these communication channels are identical for the TOE.

6.2.2.2 Cryptographic support for authentication

FCS_CKM.1/AUTH: Cryptographic key generation for authentication

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/AUTH	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **ECKA-EG with X9.63 KDF**[22] and specified cryptographic key sizes **128 bit**[23] that meet the following: **[18, chapter 8.1.2] in combination with [14] and [17]**[24].

FCS_COP.1/AUTH: Cryptographic operation for authentication

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/AUTH	The TSF shall perform **calculation of Message Authentication Code**[25] in accordance with a specified cryptographic algorithm **AES-CMAC**[26] and cryptographic key sizes **128 bit**[27] that meet the following: **[7] in combination with [27]**[28].

Application note: The TOE must fulfil the requirements FCS_CKM.1/AUTH and FCS_COP.1/AUTH for the establishment of the communication channel between itself and a remote CS according to

22 [assignment: *key generation algorithm*]

23 [assignment: *cryptographic key sizes*]

24 [assignment: *list of standards*]

25 [assignment: *list of cryptographic operations*]

26 [assignment: *cryptographic algorithm*]

27 [assignment: *cryptographic key sizes*]

28 [assignment: *list of standards*]

FTP_ITC.1 (CS) and for the establishment of the communication channel between itself and the remote storage unit according to FTP_ITC.1 (STORAGE).

6.2.2.3 Cryptographic support for signature creation and verification

FCS_COP.1/SIG: Cryptographic operation for digital signature creation

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SIG	The TSF shall perform **digital signature creation**[29] in accordance with a specified cryptographic algorithm **ECDSA**[30] and cryptographic key sizes **256 bit**[31] that meet the following: **[17]**[32].

Application note: The TOE is required to perform *digital signature creation* in case of authenticating the CS for establishing the communication channel between itself and a remote CS in FTP_ITC.1 (CS) or between itself and the remote storage unit in FTP_ITC.1 (STORAGE).

FCS_COP.1/HASH: Cryptographic operation for hashing for signature creation and verification

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HASH	The TSF shall perform **hashing for signature creation and verification** [33] in accordance with a specified cryptographic algorithm **SHA-256**[34] and cryptographic key sizes **none**[35] that meet the following: **[14]**[36].

29 [assignment: *list of cryptographic operations*]

30 [assignment: *cryptographic algorithm*]

31 [assignment: *cryptographic key sizes*]

32 [assignment: *list of standards*]

33 [assignment: *list of cryptographic operations*]

34 [assignment: *cryptographic algorithm*]

35 [assignment: *cryptographic key sizes*]

36 [assignment: *list of standards*]

6.2.2.4 Random Number Generation

FCS_RNG.1 Random number generation (Class DRG.2)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1	The TSF shall provide a *deterministic*[37] random number generator that implements: **If initialized with a random seed generated on the underlying operating system, the internal state of the RNG shall have at least 100 bit of entropy.** **The RNG provides forward secrecy.** **The RNG provides backward secrecy.**[38]
FCS_RNG.1.2	The TSF shall provide random numbers that meet **The RNG, initialized with a random seed using the Linux NPTRNG, generates output for which $k = 2^{19}$ strings of bit length 128 are mutually different with probability $k > 2^{19}$ and $\varepsilon < 2^{-10}$.** **Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A, additional test suites: none.** [39]

Application note: The TOE generates random numbers used for the authentication mechanisms used for establishing communication channels as defined in FTP_ITC.1 (CRYPTO), FTP_ITC.1 (CS), FTP_ITC.1 (STORAGE), and FTP_ITC.1 (TAPP). The TOE uses an RNG of class DRG.2 according to [22, chapter 4.7].

37 [selection: physical, non-physical true, deterministic, physical hybrid, deterministic hybrid]

38 [assignment: list of security capabilities]

39 [assignment: a defined quality metric]

6.2.3 Class FDP: User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the **TSP_ACC**[40] on
a) list of subjects: Client Software Applications
b) objects: Archive Objects
c) operations: archive requests, any other operations which are out of scope of this ST but added to a product or part of a product which claims to serve as a TOE: none[41] [42].

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the **TSP_ACC**[43] to objects based on the following:
a) list of subjects: Client Software Applications
- **Security Attribute: Client Software Application Identity**

b) objects: Archive Objects
- **Security Attribute(s): Archive Object Specific Credentials (the AOID, the retention time)**[44].

40 [assignment: *access control SFP*]

41 [assignment: *any other operations which are out of scope of this ST ~~PP~~ but added to a product or part of a product which claims to serve as a TOE*]

42 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

43 [assignment: *access control SFP*]

44 [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: • **Only an identified and authenticated CS is allowed to submit a SIP for storage.** • **Only an identified and authenticated CS which provides a valid Archive Object Specific Credential, at least the AOID, is authorized to read-out or delete the respective AIP.** • **Only an identified and authenticated CS which provides a valid Archive Object Specific Credential, at least the AOID, and a justification is authorized to delete the respective AIP before expiry of the retention time.** • **Only an identified and authenticated CS which provides a valid Archive Object Specific Credential, at least the AOID, is authorized to read-out evidence data for the respective AIP** [45] • **none** [46].
FDP_ACF.1.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**[47].
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none**[48].

Application note: These access control rules will be enforced by the ArchiSafe module. They are in addition and completely independent to access controls implemented in the CS or in the SU. **The TOE access model implements a role based access control model with one role 'Benutzer'.**[49] In all cases the access control model has to ensure that unauthorized access, e.g. between different clients with identical AOID ranges, is not possible.

45 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

46 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

47 [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]

48 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

49 [assignment: The ArchiSafe access control model can also be more complex than depicted here, e.g. group-based or role-based and may consider several clients in parallel.]

FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the **TSP_IFC**[50] on

- **Subjects: Client Software Applications, Storage Unit, another trustworthy application which connects to the Storage Unit**
- **Information: Archive Objects, Evidence Data**
- **Operations: Archive Requests**
- **list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP: none**[51] [52].

FDP_IFF.1 Simple security attributes

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

50 [assignment: *information flow control SFP*]

51 [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]

52 [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]

FDP_IFF.1.1	The TSF shall enforce the **TSP_IFC**[53] based on the following types of subject and information security attributes: • **Subject: Client Software Applications,** o **Security Attributes: Client Software Application Identity** • **Subject: Storage Unit** o **Security Attributes: Storage Unit Identity** • **Subject: another trustworthy application which connects to the Storage Unit** o **Security Attributes: another trustworthy Application Identity** • **Information: Data objects** o **Security Attributes: Type of Archive Request** • **Information: Evidence Data** o **Security Attributes: Type of Archive Request** [54] • **none** [55].
FDP_IFF.1.2	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: • **For all requests** o **The TOE must select and execute the appropriate TOE configuration data and rules based on the Client Software Application Identity and/or the archive request type.** o **The TOE does not interpret or modify any input or output data, i.e. AOIDs as well as data of SIPs or AIPs (in terms of scripts, etc.)**

53 [assignment: *information flow control SFP*]

54 [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

55 [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]

Application Note: Adding data to SIPs in accordance with configuration data or rules defined by the organization using the TOE which govern the handling of SIPs must not compromise the security objectives of this ST.

- **Archive Submission Requests**
  - **The TOE forwards the SIP to the Evidence Preservation Component, to the storage unit or to another trustworthy application which in turn forwards the SIP to the storage unit.**

    Application Note: The TOE or the IT environment needs to be configured in such a way that the immediate generation of the data for the evidence database is possible based on this information flow.
  - **If the TOE does not generate the AOID by itself, the TOE shall receive the AOID from the respective component.**
  - **The TOE shall return the AOID for each submitted submission information package to the submitting Client Software Application as result of a successful archive submission request.**

- **Archive Retrieval Requests**
  - **The TOE retrieves for each valid AOID the assigned archival information package from the storage unit.**
  - **The TOE returns for each valid AOID the assigned archival information package to the requesting Client Software Application.**

- **Archive Deletion Requests**
  - **The TOE deletes the AIP identified by the AOID from the storage unit.**
  - **The TOE returns the success of the operation to the requesting Client Software Application.**

	• **Archive Evidence Requests** o **The TOE requests Evidence Data from the Evidence Preservation Component for each AIP identified by an AOID.** o **The TOE returns the received Evidence Data to the requesting Client Software Application** [56] • **none** [57].
FDP_IFF.1.3	The TSF shall enforce the following • **The TOE has to ensure that the rules for guaranteeing the interoperability of data formats will be performed at Archive Submission or at Archive Retrieval Request** [58] • **none** [59]**.** *Application Note: The ST does not want to specify in detail at which point in time the data format will be checked. However, it shall be ensured that for each SIP the rules will be enforced.*
FDP_IFF.1.4	The TSF shall explicitly authorise an information flow based on the following rules: **none**[60].
FDP_IFF.1.5	The TSF shall explicitly deny an information flow based on the following rules: • **The TOE must not perform an archive request, if the access control rules defined in FDP_ACF.1 denies the access.** • **The TOE must not perform an archive request, if the verification procedures of the rules deposited in the TOE fail or cannot be**

56 [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

57 [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

58 [assignment: *additional information flow control SFP rules*]

59 [assignment: *additional information flow control SFP rules*]

60 [assignment: *rules, based on security attributes, that explicitly authorise information flows*]

completely executed [61]

- **none** [62].

Application Note: This SFR ensures that the ownership of an archive object will be imported from the long-term storage unit.

6.2.4 Class FIA: Identification and Authentication

FIA_UAU.2 User authentication before any action

Hierarchical to:	FIA_UAU.1	Timing of authentication
Dependencies:	FIA_UID.1	Timing of identification

FIA_UAU.2.1	The TSF shall require each Client Software Application to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that Client Software Application.

FIA_UID.2 User identification before any action

Hierarchical to:	FIA_UID.1	Timing of identification
Dependencies:	No dependencies.	

FIA_UID.2.1	The TSF shall require each Client Software Application to be successfully identified before allowing any other TSF-mediated actions on behalf of that Client Software Application.

61 [assignment: *rules, based on security attributes, that explicitly deny information flows*]

62 [assignment: *rules, based on security attributes, that explicitly deny information flows*]

6.2.5 Class FMT: Security management

FMT_MSA.1 (Access) Management of security attributes

Hierarchical to:	No other components.	
Dependencies:	[FDP_ACC.1	Subset access control, or
	FDP_IFC.1	Subset information flow control]
	FMT_SMR.1	Security roles
	FMT_SMF.1	Specification of Management Functions

FMT_MSA.1.1	The TSF shall enforce the **TSP_ACC**[63] to restrict the ability to *modify* and *delete* [64] the security attributes **access control rules**[65] to **Administrators**[66].

Application Note: It is worth to mention that the role "Administrator" **is** maintained by the TOE.[67] The term "access control rules" encompasses all rules defined by TSP_ACC **and no** additional access control rules defined by the product developer[68] or the organization using the TOE.

FMT_MSA.1 (Rules) Management of security attributes

Hierarchical to:	No other components.	
Dependencies:	[FDP_ACC.1	Subset access control, or
	FDP_IFC.1	Subset information flow control]
	FMT_SMR.1	Security roles
	FMT_SMF.1	Specification of Management Functions

FMT_MSA.1.1	The TSF shall enforce the **TSP_IFC**[69] to restrict the ability to *modify* and *delete* [70] the security attributes **TOE configuration data and rules**[71] to **Administrators**[72].

63 [assignment: *access control SFP(s), information flow control SFP(s)*]

64 [selection: *change_default, query, modify, delete, [assignment: other operations]*]

65 [assignment: *list of security attributes*]

66 [assignment: *the authorised identified roles*]

67 [assignment: *may be*]

68 [assignment: *as well as potential*]

69 [assignment: *access control SFP(s), information flow control SFP(s)*]

70 [selection: *change_default, query, modify, delete, [assignment: other operations]*]

Application Note: It is worth to mention that the role "Administrator" **is** maintained by the TOE.[73] The term "TOE configuration data and rules" encompasses all security relevant attributes which serve to confirm the identity of components connected to the TOE, allowed types of archive requests, as well as access control rules and information flow control rules.

FMT_MSA.3 (ACCESS) Static attribute initialisation

Hierarchical to:	No other components.	
Dependencies:	FMT_MSA.1	Management of security attributes
	FMT_SMR.1	Security roles

FMT_MSA.3.1	The TSF shall enforce the **TSP_ACC**[74] to provide *restrictive*[75] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow **nobody**[76] to specify alternative initial values to override the default values when an object or information is created.

Application Note: This SFR shall ensure that all security attributes relevant for accessing archive objects (e.g. the possible types of archive requests) will be initialized with secure default values and that these defaults cannot be changed.

FMT_MSA.3 (Rules) Static attribute initialisation

Hierarchical to:	No other components.	
Dependencies:	FMT_MSA.1	Management of security attributes
	FMT_SMR.1	Security roles

FMT_MSA.3.1	The TSF shall enforce the **TSP_IFC**[77] to provide *restrictive*[78] default values for security attributes that are used to enforce the SFP.

71 [assignment: *list of security attributes*]

72 [assignment: *the authorised identified roles*]

73 [assignment: *may be*]

74 [assignment: *access control SFP(s), information flow control SFP(s)*]

75 [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

76 [assignment: *the authorised identified roles*]

FMT_MSA.3.2	The TSF shall allow **nobody**[79] to specify alternative initial values to override the default values when an object or information is created.

Application Note: This SFR shall ensure that all security attributes relevant for the information flow control (e.g. the TOE configuration data and the rules for verification) will be initialized with secure default values and that these defaults cannot be changed. This holds also valid for the mandatory format verification at submission or retrieval request.

FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1	The TSF shall maintain the roles **authorized Client Software Application** [80]**, none** [81].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

Application Note: The roles "Administrator" and "Organization using the TOE" **are** defined by the operational environment and are therefore not maintained by the TSF.[82] The term "Users" denominates in a first step the different client software applications accessing the archive or vice versa an authorized Client Software Application denominates active external entities acting on behalf of an authorized user.

6.2.6 Class FPT: Protection of the TSF

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

Dependencies: No dependencies.

77 [assignment: *access control SFP(s), information flow control SFP(s)*]

78 [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

79 [assignment: *the authorised identified roles*]

80 [assignment: *the authorised identified roles*]

81 [assignment: *the authorised identified roles*]

82 [assignment: *may be, then*]

FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret **TOE configuration data, none** [83] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use **none**[84] when interpreting the TSF data from another trusted IT product.

6.2.7 Class FTP: Trusted path/channels

FTP_ITC.1 (CRYPTO) Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 (CRYPTO)	The TSF shall provide a communication channel between itself and a crypto provider[85] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and disclosure[86]. *Application note: It is worth to note, that the TOE itself is required to implement secure cryptographic operations or other measures needed to provide assured identification of the channels endpoints and to establish a trusted channel between itself and the trusted crypto provider. It is not acceptable to assume that the environment will provide this channel.*
FTP_ITC.1.2 (CRYPTO)	The TSF shall permit *the TSF* [87] to initiate communication via the trusted channel.

83 [assignment: *list of TSF data types*]

84 [assignment: *list of interpretation rules to be applied by the TSF*]

85 [refinement: *another trusted IT product*]

86 [refinement: *modification or disclosure*]

87 [selection: *the TSF, another trusted IT product*]

FTP_ITC.1.3 (CRYPTO)	The TSF shall initiate communication via the trusted channel for **performing all types of cryptographic operations apart from operations which serve to provide assured identification of endpoints between the TOE and non-TOE components as well as to protect the corresponding communication channels from modification and disclosure**[88] [89]. *Application note: Taking the upper application note into account,* ***the product developer choose*** *for*[90] ***(b) to implement secure cryptographic operations or other measures needed for assured identification of other communication endpoints as well as to protect communication channel data by the TOE itself.*** *It is worth to mention that, when using the crypto provider functionalities to assure communication endpoints and to establish trusted channels between the TOE and non-TOE components, these functionalities become virtually part of the TOE and are therefore part of a product evaluation.*

88 [refinement: *modification or disclosure*]

89 [assignment: *list of functions for which a trusted channel is required*]

90 [assignment: *product developers shall be free*][selection: *(a) using the crypto providers functionality to assure identification of other communication endpoints as well as to protect communication channel data between the TOE and other non-TOE components from modification or disclosure (see other FTP_ITC components) or,*]

Application Note: FTP_ITC.1 (CRYPTO) realizes a trusted path between the TOE and the TOE's crypto provider using TLS v1.2 according to [9] with the cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 according to [11] and the EC curve *secp256r1* according to [28] (see FCS_COP.1/TLS). *Mutual authentication* is provided through ECDSA signature verification for server authentication using SHA-256 and ECDSA signature generation for client authentication using SHA-256 according to [29] and [14]. *Session key agreement* is provided through the key derivation function PRF based on HMAC with SHA-256 (*tls_prf_sha256*) (see FCS_CKM.1/TLS) according to [24] and [14]. *Cryptographic key distribution* is provided through the TLS key exchange with ECDHE from [28] and [30]. *Protection of the payload* is provided through authenticated encryption and decryption using AES in GCM mode with a 16-octet initialization vector ICV according to [27], [31], [26], and [25].

FTP_ITC.1 (CS) Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 (CS)	The TSF shall provide a communication channel between itself and a remote Client Software Application[91] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification[92]. *Application note: It is worth to mention, that the TOE itself is required to implement secure cryptographic operations or other measures needed to provide assured identification of its endpoints and to establish a trusted channel between itself and the remote Client Software Application. It is not acceptable to assume that the environment will provide this channel.*
FTP_ITC.1.2 (CS)	The TSF shall permit *remote Client Software Application*[93] to initiate communication via the trusted channel.
FTP_ITC.1.3 (CS)	The TSF shall initiate communication via the trusted channel **only for request responses**[94].

91 [refinement: *another trusted IT product*]

92 [refinement: *modification or disclosure*]

93 [selection: *the TSF, another trusted IT product*]

94 [assignment: *list of functions for which a trusted channel is required*]


Open
Limit

FTP_ITC.1 (STORAGE) Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 (STORAGE)	The TSF shall provide a communication channel between itself and a remote storage unit[95] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification[96]. *Application note: It is worth to mention, that the TOE itself is required to implement secure cryptographic operations or other measures needed to provide assured identification of its endpoints and to establish a trusted channel between itself and the remote storage unit. It is not acceptable to assume that the environment will provide this channel.*
FTP_ITC.1.2 (STORAGE)	The TSF shall permit *the TSF* [97] to initiate communication via the trusted channel.
FTP_ITC.1.3 (STORAGE)	The TSF shall initiate communication via the trusted channel for • **Archive Retrieval Requests** • **Archive Deletion Requests** • **list of additional requests accepted by the TSF: none**[98].

95 [refinement: *another trusted IT product*]

96 [refinement: *modification or disclosure*]

97 [selection: *the TSF, another trusted IT product*]

98 [assignment: *list of additional requests accepted by the TSF*]

FTP_ITC.1 (TAPP) Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 (TAPP)	The TSF shall provide a communication channel between itself and remote trustworthy application (e. g. the Evidence Preservation Component in Figure 1)[99] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and disclosure[100]. *Application note: It is worth to mention, that the TOE itself is required to implement secure cryptographic operations or other measures needed to provide assured identification of its endpoints and to establish a trusted channel between itself and the remote storage unit. It is not acceptable to assume that the environment will provide this channel.*
FTP_ITC.1.2 (TAPP)	The TSF shall permit *the TSF* [101] to initiate communication via the trusted channel.
FTP_ITC.1.3 (TAPP)	The TSF shall initiate communication via the trusted channel for • **Archive Submission Requests** • **Archive Evidence Requests** • **list of additional requests accepted by the TSF: none**[102] .

99 [refinement: *another trusted IT product*]

100 [refinement: *modification or disclosure*]

101 [selection: *the TSF, another trusted IT product*]

102 [assignment: *list of additional requests accepted by the TSF*]

6.3 Security Assurance Requirements (SARs)

The security assurance requirements for the evaluation of the TOE and its development and operating environment are those taken from the

Evaluation Assurance Level 4 (EAL 4).

augmented by the following component:

ALC_FLR.1.

The following “Table 3” gives an overview on the security assurance requirements that have to be fulfilled by the TOE.

Assurance class	**Assurance components**
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Live-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_FLR.1 Basic flaw remediation
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
ASE: Security target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition

Assurance class	Assurance components
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis

Table 3: TOE Security Assurance Requirements

6.4 Rationale for the Security Functional Requirements

The following table indicates that the security objectives pointed out in section 4.1 will be covered by the security functional requirements represented in section 6.2 of this Security Target.

	O.ACCESS	O.AOID	O.AUTH_REQUEST	O.CONFIGURATION	O.CRYPTO_SPOOF	O.DATA_EXAM	O.DELETION	O.DELETION_LOG	O.RETURN	O.STORAGE_SPOOF	O.TOE_AUTHENT	O.TOE_COMM
FAU_GEN.1								X				
FCS_CKM.1/AUTH										X	X	X
FCS_CKM.1/TLS					X						X	X
FCS_COP.1/AUTH										X	X	X
FCS_COP.1/SIG										X	X	X
FCS_COP.1/TLS					X						X	X
FCS_COP.1/HASH										X	X	X
FCS_RNG.1										X	X	X
FDP_ACC.1	X		X				X					
FDP_ACF.1	X		X				X					
FDP_IFC.1		X		X		X			X	X		
FDP_IFF.1		X		X		X			X	X		
FIA_UAU.2			X									
FIA_UID.2			X									
FMT_MSA.1 (Access)			X									
FMT_MSA.1 (Rules)				X								
FMT_MSA.3 (Access)			X									
FMT_MSA.3 (Rules)				X		X						
FMT_SMR.1			X									
FPT_TDC.1				X								
FTP_ITC.1 (CRYPTO)					X						X	X
FTP_ITC.1 (CS)											X	X
FTP_ITC.1 (STORAGE)										X	X	X
FTP_ITC.1 (TAPP)										X	X	X

Table 4: Coverage of the Security Objectives by Security Functional Requirements

O.ACCESS: FDP_ACF.1 and FDP_ACC.1 guarantee that the TOE will only allow the specified types of archive requests.

O.AOID: The rules enforced by FDP_IFC.1 and FDP_IFF.1 ensure that the TOE does not interpret any input or output parameters in terms of a script and that the TOE does not change these values. This holds also valid for the AOID.

O.AUTH_REQUEST: FDP_ACC.1 and FDP_ACF.1 enforces the actual access control based on credentials. FIA_UAU.2 and FIA_UID.2 deliver the credential "client application identity" for the access control mechanism. FMT_MSA.1 (Access) and FMT_MSA.3 (Access) ensure that the access control defaults are set restrictive and that this default cannot be changed. FMT_SMR.1 ensures that the TOE is able to manage a role for the authenticated client applications.

O.CONFIGURATION: FDP_IFC.1 and FDP_IFF.1 ensures that the right configuration data will be selected and executed. This includes also the denial of an access in case of incomplete or not successful performance of the rules. FMT_MSA.1 (Rules) and FMT_MSA.3 (Rules) ensures that there are restrictive defaults for the configuration data and that these defaults cannot be changed. FPT_TDC.1 ensures that the configuration rules will be interpreted correctly by the TOE.

O.CRYPTO_SPOOF: FCS_CKM.1/TLS, FCS_COP.1/TLS, and FTP_ITC.1 (CRYPTO) enforce a reliable identification of a dedicated crypto provider. Thus, the selected (defined) trustworthy crypto provider cannot be substituted unnoticed.

O.DATA_EXAM: FDP_IFC.1 and FDP_IFF.1 enforce the verification of SIPs/AIPs at the point of submission or at the point of retrieval. FMT_MSA.3 (Rules) ensures that there are restrictive defaults for this and that nobody can change these defaults. FMT_MSA.1 (Rules) is not relevant here.

O.DELETION: FDP_ACC.1 and FDP_ACF.1 enforce that nobody will be able to delete an archive object before the expiry of its retention time without any justification.

O.DELETION_LOG: FAU_GEN.1 guarantees that any erasure request to archive objects before the expiry of their retention time will be recorded including the justification for that activity.

O.RETURN: FDP_IFC.1 and FDP_IFF.1 enforce that after successful storage of a data object the TOE returns the archive object ID (AOID) to the submitting client software application.

O.STORAGE_SPOOF: FDP_IFC.1 and FDP_IFF.1 ensure that SIPs intended to be stored will be forwarded to the SU or another trustworthy application. FCS_CKM.1/AUTH, FCS_COP.1/AUTH, FCS_COP.1/SIG, FCS_COP.1/HASH, FCS_RNG.1, FTP_ITC.1 (STORAGE), and FTP_ITC.1 (TAPP) ensure that the SU and the other trustworthy application will be identified and authenticated before it will be used by the TOE and can therefore not be replaced without notice.

O.TOE_AUTHENT: FCS_CKM.1/AUTH, FCS_COP.1/AUTH, FCS_COP.1/SIG, FCS_COP.1/HASH, FCS_CKM.1/TLS, FCS_COP.1/TLS, FCS_RNG.1, FTP_ITC.1 (CRYPTO), FTP_ITC.1 (CS), FTP_ITC.1 (STORAGE) and FTP_ITC.1 (TAPP) require a mutual authentication of the end points of the respective communication connections. This also includes the authentication of the TOE against all the other end points, namely the client software application, the crypto provider, the storage unit and other trustworthy application (e.g. the Evidence Preservation Component).

O.TOE_COMM: FCS_CKM.1/AUTH, FCS_COP.1/AUTH, FCS_COP.1/SIG, FCS_COP.1/HASH, FCS_CKM.1/TLS, FCS_COP.1/TLS, FCS_RNG.1, FTP_ITC.1 (CRYPTO), FTP_ITC.1 (CS), FTP_ITC.1 (STORAGE) and FTP_ITC.1 (TAPP) require the protection of the communication against modification, namely the communication with the client software application, the crypto provider, the storage unit and other trustworthy application (e.g. the Evidence Preservation Component).

6.5 Rationale for the Security Assurance Requirements

The evaluation assurance level EAL4 with augmentations chosen in this security target permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources.

The selection of the component **ALC_FLR.1** requiring the TOE developer to track and correct flaws in the TOE subsequently provides assurance that the TOE will be maintained and supported in the future.

All dependencies resulting directly or indirectly from the augmentation ALC_FLR.1 are discussed in the following:

The component **ALC_FLR.1** has no dependencies.

6.6 Rationale for the Security Functional Requirements and their dependencies

The following table shows the security functional requirements of the TOE, their dependencies and how these dependencies are resolved.

SFR	Dependencies	Resolved
FAU_GEN.1	FPT_STM.1	Resolved by the TOE environment
FCS_CKM.1/AUTH	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4	Resolved by FCS_COP.1/AUTH Not resolved (see justification 2)
FCS_CKM.1/TLS	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4	Resolved by FCS_COP.1/TLS Not resolved (see justification 2)
FCS_COP.1/AUTH	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	Resolved by FCS_CKM.1/AUTH Not resolved (see justification 2)
FCS_COP.1/SIG	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	Not resolved (see justification 2)
FCS_COP.1/TLS	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	Resolved by FCS_CKM.1/TLS Not resolved (see justification 2)
FCS_COP.1/HASH	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	Not resolved (see justification 1) Not resolved (see justification 2)
FCS_RNG.1	No dependencies	---
FDP_ACC.1	FDP_ACF.1	Resolved
FDP_ACF.1	FDP_ACC.1	Resolved
	FMT_MSA.3	Resolved by FMT_MSA.3 (Access)
FDP_IFC.1	FDP_IFF.1	Resolved
FDP_IFF.1	FDP_IFC.1	Resolved
	FMT_MSA.3	Resolved by FMT_MSA.3 (Rules)
FIA_UAU.2	FIA_UID.1	Resolved by hierarchical FIA_UID.2
FIA_UID.2	No dependencies	---
FMT_MSA.1 (Access)	FDP_ACC.1 or FDP_IFC.1	Resolved by FDP_ACC.1
	FMT_SMF.1	Not resolved because the TOE does not have management functions.
	FMT_SMR.1	Not resolved because the role “Administrator” is assumed to be managed by the IT environment.
FMT_MSA.1 (Rules)	FDP_ACC.1 or FDP_IFC.1	Resolved by FDP_IFC.1
	FMT_SMF.1	Not resolved because the management of these security attributes is out of scope.
	FMT_SMR.1	Not resolved because the role “Administrator” is assumed to be managed by the IT environment.
FMT_MSA.3 (Access)	FMT_MSA.1	Resolved by FMT_MSA.1 (Access)
	FMT_SMR.1	Not resolved because the role “Administrator” is assumed to be managed by the IT environment.

SFR	Dependencies	Resolved
FMT_MSA.3 (Rules)	FMT_MSA.1	Resolved by FMT_MSA.1 (Rules)
	FMT_SMR.1	Not resolved because the role “Administrator” is assumed to be managed by the IT environment.
FMT_SMR.1	FIA_UID.1	Resolved by hierarchical FIA_UID.2
FPT_TDC.1	No dependencies	---
FTP_ITC.1 (CRYPTO)	No dependencies	---
FTP_ITC.1 (CS)	No dependencies	---
FTP_ITC.1 (STORAGE)	No dependencies	---
FTP_ITC.1 (TAPP)	No dependencies	---

Table 5: Rationale for the Security Functional Requirements and their dependencies

Justification for missing dependencies

Justification 1: The hash algorithm as defined in FCS_COP.1/HASH does not need any key material. As such the dependency to an import or generation of key material as well as the dependency to a deletion of key material is omitted for this SFR.

Justification 2: The key material used in FCS_CKM.1/AUTH, FCS_CKM.1/TLS, FCS_COP.1/AUTH, FCS_COP.1/SIG, FCS_COP.1/TLS, and FCS_COP.1/HASH does not need cryptographic key destruction because this ephemeral key material is only kept in memory of the machine running the TOE and automatically lost when the TOE stops operation.

7 TOE Summary Specification

The following paragraph provides a TOE summary specification describing how the TOE meets each SFR.

7.1 SF 1: Secure Client TOE Access

The TOE controls the access to the archive, permitting archive requests only from successfully authenticated CS. Therefore the TOE owns a reliable identification and authentication process using a single-instance of the TOE component *Credential Store* conducting the identification and authentication.

During installation of the TOE at the customer's location, the company running the TOE installation in secure environment has to generate an asymmetric *secp256r1* ECC key pair according to [12] and has to derive from this a self-signed X.509 certificate according to [10]. The generated cryptographic key material must be securely imported into the TOE by the trustworthy administrator of each startup of a TOE installation.

On the other hand, the cryptographic key material to be used by a CS instance has to be generated in an analogue way leading to the *secp256r1* ECC key pair of the CS. Again the public key is packed into a self-signed X.509 certificate. The public key of this certificate has to be configured in the TOE by the trustworthy administrator of the TOE installation for each CS instance the TOE should permit access.

The identification and authentication of a CS is done via the public key of this X.509 certificate and the so-called *IdentToken*. Thus, for the first registration of a CS, the following procedure takes place: the CS sends a unique token, the *IdentToken*, to the TOE. The TOE checks that the CS has been configured by the TOE administrator by grabbing for the public key of the CS, otherwise the TOE access will not be granted for the CS. In case of a configured CS, the TOE derives the key K_{mac} from a newly generated ephemeral private key and the static public key of the CS using ECKA-EG according to [18, chapter 8.1.2] and the X9.63 key derivation function according to [17] with the SHA-256 digesting algorithm according to [14].

In the next step, the TOE sends his ephemeral public key to the CS - along with the plain binary ECDSA signature S over this ephemeral public key using his private ECC key. The CS verifies the signature using the TOE's public key proving the TOE's authenticity. If successful, the CS calculates the key K_{mac}' from the TOE's ephemeral public key and his own private ECC key once again using ECKA-EG and the X9.63 key derivation function according to [18, chapter 8.1.2] and [17].

After the successful establishment of these key agreement procedures, for each message the CS wants to send, the CS calculates the message authentication code MAC' using his key K_{mac}' and the AES-CMAC-128 algorithm according to [7]. The TOE tries to verify the message's authenticity through calculation of the message authentication code MAC using his key K_{mac}. In the case that MAC meets MAC', the message sent is successfully authenticated; otherwise the procedure is cancelled and this security incidence is audited by the TOE. For the TOE response, the same procedure is used vice

versa (FCS_CKM.1/AUTH, FCS_COP.1/AUTH, FCS_COP.1/SIG, FCS_COP.1/HASH, FCS_RNG.1, FDP_ACC.1, FDP_ACF.1, FIA_UAU.2, FIA_UID.2, FMT_MSA.1 (Access), FMT_MSA.3 (Access), FMT_SMR.1, and FTP_ITC.1 (CS)). The generated key material of the TOE is included in the PKCS#12 file *AuthToken.p12*.

For X.509 certificates used in this procedure, neither the validity nor the revocation of the certificate will be checked by the TOE. The TOE checks the correctness of the X.509 certificate by checking its structural correctness.

The communication via this trusted channel is always initiated by the CS (FDP_IFC.1).

If the request of a client could not be successfully authenticated by the TOE, the request is rejected. If the request of a client is successfully identified and authenticated, the TOE generates a so-called *session ID* for the CS returning this session ID to the CS which should store this session ID during the session. In every further archive request, the client has to use his session ID ensuring that the intended configuration of the Crypto Provider component is used by the CS. A client can request more than one session ID. All unsuccessful client requests are audited by the TOE (FAU_GEN.1).

If the TOE receives a successfully authenticated submission archive request, the CS identity – represented by the so-called *IdentToken* - will be closely connected to the submitted data objects. Thus, the TOE augments the data objects to be archived with the ID of the submitting client software application.

Access to archive objects will only be granted to this particular CS which has submitted the data object for archiving (FDP_ACC.1, FDP_ACF.1, FMT_MSA.1 (Rules), FMT_MSA.1 (Access), FMT_MSA.3 (Rules), and FMT_MSA.3 (Access)). Therefore the CS must be successfully identified by the submitted *IdentToken* and the message of the CS must be successfully authenticated through submitting the correct CMAC.

Before being successfully authenticated and identified, the TOE doesn’t allow any archive requests to a CS (FIA_UAU.2, FIA_UID.2, FDP_IFC.1, FDP_IFF.1, and FMT_MSA.1 (Rules)).

For a successfully identified and authenticated CS the TOE allows the following request types (FDP_IFF.1, FMT_MSA.1 (Rules), and FMT_MSA.3 (Rules)):

- Request for storing data objects in the storage system
- Request for retrieving data objects from the storage system
- Request for erasing data objects from the storage system
- Request for retrieving evidence records
- Request for reading meta information

As the result of an archive request the TOE returns at least the OID and the AOID having received from the SU to the submitting CS without interpretations (FDP_IFC.1, FDP_IFF.1) providing an unambiguous association between the archive request and the meta information of the AIP as proof of the integrity and authenticity of the archive request and as proof of the identity of the owner of the AIP (FMT_MSA.1 (Rules), and FMT_MSA.3 (Rules)).

7.2 SF 2: Data Object Verification

The TOE prevents the storage of invalid data objects by reliable verification of the submitted data objects before forwarding them to the SU (or another trusted application which in turn forwards the SIP's to the SU) through the following measures:

If a CS requests the TOE for submission of a data object to a dedicated archive, the TOE validates this data object taking in account the rules defined by the administrator of the TOE used for creation of the data object which the CS has submitted along with this submission archive request.

The rules defined by the administrator of the TOE can only be loaded into the TOE through the use of the function *ol_addProfile*. This function can only be used by a successfully authenticated user. The TOE only imports these filter rules if the verification of the submitted data and the according *ProfileIdentToken* succeeds. For any further archive submission request made by a TOE user, this *ProfileIdentToken* has to be part of the request so that the TOE can ensure the interrelation between the archive request and the corresponding filter rules.

If the TOE imports filter rules, the TOE checks if the data of the rule are syntactically correct taking into account the XML schema associated with the *ProfileIdentToken*. The TOE refuses to import any filter rules not being syntactically upright.

For the submission of data objects, the CS must submit his *ProfileIdentToken*. The TOE is capable to consistently interpret these XML schemas as part of the TOE configuration data being shared with the underlying system (FPT_TDC.1). If the TOE could not locate the XML schema named in this submission archive request in the TOE's client-dependant configuration, an error message is returned to the CS and the submission archive request is cancelled (FDP_IFC.1, FDP_IFF.1) and an audit record is generated by the TOE (FAU_GEN.1). If the data object to be archived doesn't contain at least an OID and retention time as meta information of the data object (as part of the request or as part of the imported filter rules), the request is rejected and an error message is sent to the CS.

If the TOE has successfully located the XML schema named in the submission archive request of the CS, the TOE uses this XML schema for syntactical validation of the SIP. If the validation of the SIP against the XML schema fails, an error message is returned to the CS, the submission archive request is cancelled and the unsuccessful verification of the XML schema is audited by the TOE (FAU_GEN.1).

If the validation of the SIP against the XML schema succeeds, the TOE proceeds with the process of the cryptographic verification of the digital signatures contained in the SIP if applicable using the defined external Crypto Provider component.

Therefore, the SIP as an XML container is parsed with a set of XPath expressions [16]. If the XML container contains any digitally signed objects referenced by XPath expressions, the signatures and their corresponding documents are extracted from the SIP as the TOE user has assigned the TOE through the submission of the XML filter to be used for the archive request. Each extracted digital signature and its corresponding document will be cryptographically verified by the defined external Crypto Provider component. The TOE receives the overall verification result as well as an according XML verification report from the defined Crypto Provider. If any digitally signed object cannot be successfully be verified, the TOE cancels the submission request, an error message is returned to the CS and the unsuccessful verification of the digitally signed object is audited by the TOE (FAU_GEN.1).

For the communication with the defined external Crypto Provider, the TOE uses a socket-based communication channel according to SOAP v.1.1 [21] being logically distinct from other communication channels and providing assured identification of its end points and protection of the channel data from modification or disclosure. This is accomplished through a TLS v.1.2 protected communication channel according to [9] using the ECC key pair and X.509 certificate configured by the TOE administrator (FCS_CKM.1/TLS, FCS_COP.1/TLS, and FTP_ITC.1 (CRYPTO)).

After being successfully authenticated for performing all further types of cryptographic operations, the TOE initiates the communication via the mentioned trusted channel to the defined Crypto Provider.

After having successfully checked the archive request and having successfully verified all digitally signed objects being part of the SIP, the TOE applies the rules being specified by the organization using the TOE according to the context of the respective archive request in the defined order (FDP_IFC.1, and FDP_IFF.1).

The TOE protects the evidential integrity and authenticity of the content of the archived information objects by strong cryptographic operations, like digital signatures and digital time-stamps, provided by the Crypto Provider via the mentioned trusted channel. The TOE is able to sustain the non-repudiation of cryptographically signed and archived information objects by evidential proof and renewal of the electronic signatures for which the TOE uses the Crypto Provider, too. So, in other terms, the non-TOE component acting as the TOE's Crypto Provider acts as the TOE's Evidence Preservation Component, too. For this evidence renewal operation, the TOE uses the TLS v.1.2 protected communication channel as described above (FCS_CKM.1/TLS, FCS_COP.1/TLS, FCS_RNG.1, and FTP_ITC.1 (TAPP)). Because the TOE's Crypto Provider Component and the TOE's Evidence Preservation Component are united in one single external component, for the TOE the requirements FTP_ITC.1 (TAPP) and FTP_ITC.1 (CRYPTO) are identical.

7.3 SF 3: Secure Storage Unit Access[103]

If the request for submission of an SIP to the archive is successfully authenticated by the TOE and the data objects are successfully checked and verified by the TOE, the TOE immediately passes the data objects to be archived to the SU (or a trusted application which in turn passes the data objects to the dedicated SU) (FDP_IFC.1, FDP_IFF.1).

To ensure that the data objects are correctly submitted to the SU (or a trusted application which in turn submits the data objects to the dedicated SU), the TOE uses a communication channel to a trusted application which in turn submits the data objects to the dedicated SU being logically distinct from other communication channels and providing assured identification of its end points and protection of the channel data from modification or disclosure. The mutual authentication mechanisms used for providing assured end point identification are comparable to the mechanisms used for the mutual authentication of the CS described above, with the differences that there is only one instance of storage system and that the direction of authentication is inverted so that the TOE initiates the authentication process according to FTP_ITC.1 (STORAGE) sending the signed ephemeral key to the dedicated SU (FCS_CKM.1/AUTH, FCS_COP.1/AUTH, FCS_COP.1/HASH, FCS_COP.1/SIG, FCS_RNG.1, and FTP_ITC.1 (STORAGE)).

For the submission of an SIP to the archive the TOE will initiate the communication via the mentioned trusted channel.

The dedicated storage unit receives the submitted SIP from the TOE for saving and must send back a unique archive object identifier (AOID) to the TOE in case of the successful storage of the AIP. The TOE returns the AOID without interpretations to the submitting CS (FDP_IFC.1, FDP_IFF.1). Nobody is allowed to modify or delete those security attributes (FMT_MSA.1 (Rules), FMT_MSA.3 (Rules)). The SIP is now called archival information package (AIP). If the storage fails, the TOE cancels the submission request and returns an error message to the CS.

103 The security functionality SF3 and SF5 described in Chapter 1.3.1 both are handled in this chapter named ‘SF 3’. The TOE is implemented in such a way, that storing and retrieval to and from the SU use both the same security mechanisms.

If a request for retrieval of an AIP or for retrieval of the meta data of an AIP is sent to the TOE by an CS and the CS is successfully authenticated according to its authentication credentials, the retrieval archive request must include the *IdentToken* of the CS and the AOID of the AIP requested proving the ownership of the requesting CS (FDP_IFC.1, FDP_IFF.1, FMT_MSA.1 (Access), FMT_MSA.3 (Access), and FMT_SMR.1).

7.4 SF 4: Invalid Archival Information Package Erasure Prevention

The TOE prevents the erasure of AIP's by any other CS than the CS which has submitted this AIP and the erasure of AIP's before expiry of their retention time without a justification (FDP_ACC.1, FDP_ACF.1, FDP_IFC.1, FDP_IFF.1) through the following measures.

For the erasure of an archive object from the storage unit the CS has to submit an archive request for deletion of an AIP which can be successfully authenticated along with the identification data in form of the *IdentToken* which should prove that the CS submitting the request is the CS which has initially submitted the AIP. **If the AIP shall be deleted before expiration of the AIP's retention time, additionally a justification must be submitted.**

The TOE ensures that the identification data - together with the AOID of the archive object requested for deletion - are submitted to the storage unit (or a trusted application which in turn submits the request to the dedicated SU) without modification (FDP_ACC.1, FDP_ACF.1, FMT_MSA.1 (Access), FMT_MSA.3 (Access), and FMT_SMR.1).

The TOE requests the retention time for the erasure archive request by passing the AOID to the SU (or a trusted application which in turn returns the retention time to the TOE). The AIP's preservation time is, if present, the AIP's explicit ExpirationDate or the time when the AIP's retention period is reached beginning with the time of the AIP's successful submission to the SU.

The TOE enforces the following rules:

- If the expiration time is behind the current time, the TOE sends the request for deletion to the SU (or a trusted application which in turn sends the request to the SU).

- If the expiration time is before the current time, the TOE checks, if the erasure archive request includes a justification. If the erasure archive request includes a justification, the TOE initiates the deletion of the AIP through the SU (or a trusted application which in turn forwards the initiation for deletion to the dedicated SU).
- If the expiration time is before the current time and the deletion archive request doesn't include a justification, the TOE cancels the deletion archive request and an error message is returned to the CS.

For all erasure archive requests for AIP's whose expiration time is before the current time the TOE generates an audit record, regardless of the outcome of the archive request (FAU_GEN.1).

7.5 TSS Rationale

The following table shows the correspondence analysis for the described TOE security functionalities and the security functional requirements.

SFR	**SF 1**: Secure Client TOE Access	**SF 2**: Data Object Verification	**SF 3**: Secure Storage Unit Access	**SF 4**: Invalid AIP Erasure Prevention
FAU_GEN.1	X	X		X
FCS_CKM.1/AUTH	X		X	
FCS_COP.1/AUTH	X		X	
FCS_COP.1/SIG	X		X	
FCS_COP.1/HASH	X		X	
FCS_CKM.1/TLS		X		
FCS_COP.1/TLS		X		
FCS_RNG.1	X	X	X	
FDP_ACC.1	X			X
FDP_ACF.1	X			X
FDP_IFC.1	X	X	X	X
FDP_IFF.1	X	X	X	X
FIA_UAU.2	X			
FIA_UID.2	X			
FMT_MSA.1 (Access)	X		X	X
FMT_MSA.1 (Rules)	X		X	
FMT_MSA.3 (Access)	X		X	X
FMT_MSA.3 (Rules)	X		X	
FMT_SMR.1	X		X	X
FPT_TDC.1		X		
FTP_ITC.1 (CRYPTO)		X		
FTP_ITC.1 (CS)	X			
FTP_ITC.1 (STORAGE)			X	
FTP_ITC.1 (TAPP)		X		

Table 6: Rationale for the SFR and the TOE Security Functionalities

8 Acronyms

AIP	Archival information package
AOID	Archive Object Identifier
CC	Common Criteria for IT Security Evaluation
CS	Client Software Application
EAL	Evaluation Assurance Level
IT	Information Technology
OID	Object Identifier
OSP	Organisational Security Policies
PP	Protection Profile
SIP	Submission information package
SFP	Security Function Policy
ST	Security Target
SU	(Long-Term) Storage Unit
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy

9 Literature

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, September 2012, version 3.1, Revision 4, CCMB-2012-09-001, http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R4.pdf

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements, September 2012, version 3.1, Revision 4, CCMB-2012-09-002, http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R4.pdf

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements, September 2012, version 3.1, Revision 4, CCMB-2012-09-003, http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R4.pdf

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; CCMB-2012-09-004, Version 3.1, Revision 4, September 2012

[5] Common Criteria Protection Profile for an ArchiSafe Compliant Middleware for Enabling the Long-Term Preservation of Electronic Documents (ACM_PP), BSI-CC-PP-0049-2014, Version 1.2, 28.03.2014, Dr. Wolf Zimmer, Bundesamt für Sicherheit in der Informationstechnik

[6] IETF RFC 2631, Diffie-Hellman Key Agreement Method, E. Rescorla, June 1999, http://www.ietf.org/rfc/rfc2631.txt

[7] IETF RFC 4493, J. H. Song, J. Lee, T. Iwata, The AES-CMAC Algorithm, June 2006, http://www.ietf.org/rfc/rfc4493.txt

[8] IETF RFC 4998, Evidence Record Syntax (ERS), T. Gondrom, R. Brandner, U. Pordesch, http://www.ietf.org/rfc/rfc4998.txt

[9] IETF RFC 5246, T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, August 2008, http://www.ietf.org/rfc/rfc5246.txt

[10] IETF RFC 5280, Internet X.509 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile, R. Housley et al., May 2008, http://www.ietf.org/rfc/rfc5280.txt

[11] IETF RFC 5289, TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), E. Rescorla, RTFM, Inc., August 2008, http://www.ietf.org/rfc/rfc5289.txt

[12] IETF RFC 5114, Additional Diffie-Hellman Groups for Use with IETF Standards, M. Lepinski, S. Kent, January 2008, http://www.ietf.org/rfc/rfc5114.txt

[13] ISO 19005-1, Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1), 2005

[14] NIST, FIPS 180-4, Secure Hash Standard, 2012

[15] NIST Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, M. Dworkin, November 2007, http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

[16] XML Path Language (XPath) 2.0, W3C Recommendation 23rd January 2007, http://www.w3.org/TR/2007/REC-xpath20-20070123/

[17] Technische Richtlinie BSI-TR-03111, Elliptic Curve Cryptography (ECC), Version 2.0, 2012, Bundesamt für Sicherheit in der Informationstechnik, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03111/BSI-TR-03111_pdf.pdf?__blob=publicationFile

[18] Technische Richtlinie BSI-TR-03116-3, Kryptographische Vorgaben für Projekte der Bundesregierung, Teil 3 - Intelligente Messsysteme, Stand 2016, Bundesamt für Sicherheit in der Informationstechnik, 04.04.2016, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03116/BSI-TR-03116-3.pdf?__blob=publicationFile

[19] Technische Richtlinie BSI-TR-03125, Beweiswerterhaltung kryptographisch signierter Dokumente, Version 1.2, 19.12.2014, Bundesamt für Sicherheit in der Informationstechnik, https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03125/index_htm.html

[20] Einheitliche Spezifizierung der Einsatzbedingungen für Signaturanwendungskomponenten, Version 1.4, announced 19.07.2005, http://www.bundesnetzagentur.de/cae/servlet/contentblob/37092/publicationFile/2443/SpezifiziergEinsatzBedinggnId2648pdf.pdf

[21] Simple Object Access Protocol (SOAP) 1.1, W3C Note, 08.05.2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508

[22] W. Killmann, W. Schindler, „A proposal for: Functionality classes for random number generators“, Version 2.0, September 18, 2011, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf

[23] Administrationshandbuch, OpenLimit Middleware Version 3 Server, Produktversion: 1.6, Stand: 13.02.2018, Dokumentenversion: 1.35, OpenLimit SignCubes AG

[24] IETF RFC 2104, HMAC: Keyed-Hashing for Message Authentication, H. Krawczyk, M. Bellare, R. Canetti, February 1997, http://www.ietf.org/rfc/rfc2104.txt

[25] IETF RFC 5116, An Interface and Algorithms for Authenticated Encryption, D. McGrew, January 2008, http://www.ietf.org/rfc/rfc5116.txt

[26] IETF RFC 5288, AES Galois Counter Mode (GCM) Cipher Suites for TLS, J. Salowey, A. Choudhury, D. McGrew, August 2008, http://www.ietf.org/rfc/rfc5288.txt

[27] NIST, FIPS 197, Advances Encryption Standard (AES), 2001

[28] IETF RFC 4492, Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS), S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, B. Moeller, May 2006, http://www.ietf.org/rfc/rfc4492.txt

[29] ANS X9.62–2005, Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA), November 16, 2005, American National Standard for Financial Services

[30] Handbook of Applied Cryptography. A. Menezes, P. van Oorschot und O. Vanstone, CRCPress, 1996

[31] NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, E. Barker, J. Kelsey, June 2015, http://dx.doi.org/10.6028/NIST.SP.800-90Ar1

Appendix A: Cryptographic Functionality Overview

No.	Purpose	Cryptographic Mechanism	Standard of Implementation	Key Size (in Bits)	Comment
1	Authenticity	ECDSA	TR-03111	256	FCS_COP.1/SIG: identification and authentication of SU, CS, and crypto provider
2	Integrity	SHA	FIPS 180-4	256	FCS_COP.1/HASH: protection of communication with CS, crypto provider, and SU against modification
3	Confidentiality Integrity	TLS v.1.2 with cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	RFC 5246, RFC 5289	128	FCS_COP.1/TLS
4	Confidentiality Integrity	AES-CMAC	RFC 4493, NIST SP 800-38D	128	FCS_COP.1/AUTH
5	Key Agreement	ECKA-EG with X9.63 KDF	TR-03116-3, TR-03111, FIPS 180-4	128	Key Generation for FCS_COP.1/AUTH
6	Confidentiality Integrity	FTP_ITC.1 (CRYPTO): Trusted communication channel between TOE and crypto provider using FCS_COP.1/TLS			

7	Confidentiality Integrity	FTP_ITC.1 (CS): Trusted communication channel between TOE and CS using FCS_COP.1/AUTH			
8	Confidentiality Integrity	FTP_ITC.1 (STORAGE): Trusted communication channel between TOE and SU using FCS_COP.1/AUTH			
9	Confidentiality Integrity	FTP_ITC.1 (TAPP): Trusted communication channel between TOE and TAPP using FCS_COP.1/AUTH			
10	Random Number Generator	NPTRNG	AIS 31	at least 100 bit of entropy and forward and backward secrecy	FCS_RNG.1: DRG.2 for TOE identy generation

Table 7: TOE Cryptographic Functionality